Practical Macrostate Data Clustering

Brian White*

Computer Systems Laboratory

Dept. of Electrical and Computer Engineering

Cornell University

Ithaca, NY 14853

David Shalloway†

Biophysics Program

Dept. of Molecular Biology and Genetics

Cornell University

Ithaca, NY 14853

Abstract

Spectral clustering methods have been shown to outperform traditional distance-based approaches, such as k-means and hierarchical clustering, based on their use of global information encoded in eigenvectors of a matrix describing inter-item relations. Macrostate data clustering [Korenblum and Shalloway, *Phys. Rev. E*, Volume 67, 2003] used an analogy to the dynamic coarse-graining of a stochastic system to construct a linear combination of eigenvectors that probabilistically assigned items to clusters. A "minimum uncertainty criterion" lead to an objective function that minimized the inherent fuzziness of the cluster assignments. The resulting non-linear optimization problem was solved by a brute-force technique that was unlikely to scale to problems larger than a few hundred items. A novel approach to solving this optimization problem is presented. It scales to 20,000 items—the memory limitations of a commodity computational node and within range of problem sizes of biological interest. To further accommodate biological applications, the theory is amended to apply to asymmetric dissimilarity matrices, such as those derived from DNA sequence alignment scores, and the algorithm is extended to recursively examine hierarchical substructure, such as that arising during protein classification. Potential application to molecular dynamics and protein classification is discussed.

*Electronic address: bwhite@csl.cornell.edu

[†]Electronic address: dis2@cornell.edu

I. INTRODUCTION

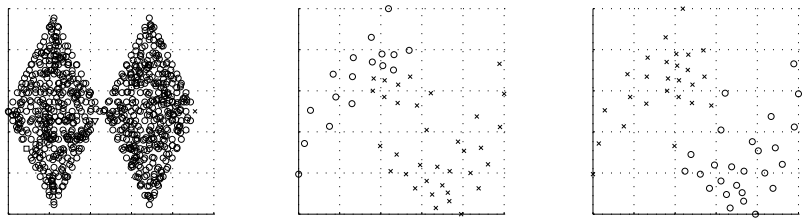
The need to coarse grain a large set of *items* to a smaller set of clusters is a ubiquitous problem in engineering and the sciences. Formally, a solution assigns N items embedded in a N_M-dimensional space to a set of m clusters, with $m \ll N$. Clustering proceeds from an $N \times N$ *dissimilarity matrix* D, where the off-diagonal element D_{ij} provides an inverse indicator of the correlations between the measurements of items i and j. The resulting *assignment functions* $\boldsymbol{w}_\alpha$ have elements $\boldsymbol{w}_\alpha(i)$ that describe the probability of item i being assigned to cluster α. If the $\boldsymbol{w}_\alpha(i)$ are continuous in the range zero to one, they describe a *fuzzy clustering*; frequently the assignment probabilities are restricted to binary values and instead give a *hard clustering*.

The dissimilarity matrix may be defined directly and externally from the clustering algorithm, as when clustering protein sequences using inter-item sequence alignment scores. Alternately, when the data are numerical, the dissimilarity matrix may be derived from the $N \times N_M$ *measurement matrix* X and a distance measure defined on the measurement space, as in

$$D_{ij} = \left[\sum_{a,b=1}^{N_M} (X_{ia} - X_{ja}) g_{ab} (X_{ib} - X_{jb}) \right]^{1/2} , \tag{1}$$

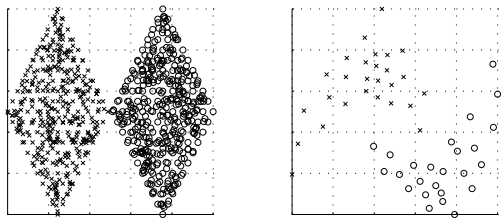
where g is a problem-specific Euclidean metric tensor. The metric tensor is taken to be the identity when the dimensions are orthogonal, but may otherwise be adjusted to account for correlations. Here, each dimension provides one set of the N_M measurements on the items. For example, in the context of a DNA microarray gene expression analysis, the items would represent genes and the measurements might correspond to gene expression levels measured at different times or across different conditions.

Clustering is an inherently global problem whose optimal solution informally maximizes each cluster's internal cohesion and external isolation [1]. Traditional distance-based methods, such as hierarchical clustering and k-means, approximate this goal by minimizing an aggregate statistic over item-item or item-cluster distances. For example, Single Linkage and Complete Linkage are agglomerative methods that iteratively merge the pair of clusters having minimum inter-cluster distance. Single Linkage defines inter-cluster distance as the minimum distance between items in different clusters, whereas Complete Linkage uses the



(a) (b) (c)

FIG. 1: Defects in distance-based methods. (a) Single Linkage applied to Two Diamonds. (b) Complete Linkage applied to Crescentric. (c) k-means applied to Crescentric.



(a) (b)

FIG. 2: Macrostate data clustering applied to (a) Two Diamonds and (b) Crescentric.

maximum distance between items in different clusters. k-means is a partitional, combinatoric method that adjusts k centroids so as to minimize the sum of the distances between each item and its respective, nearest centroid.

The notions of cohesion and isolation that intuitively define a quality clustering are inherently global connectivity relations. The sub-optimal clusterings of traditional methods, such as those in Figure 1, are often related to their inability to infer such connectivity information from inter-item distances. Single Linkage improperly clusters the Two Diamonds data set [2] in Figure 1(a) by prematurely merging two sub-clusters that straddle the interface between the diamonds. This effectively merges the two dominant clusters before integrating three singletons into their respective diamonds. Though the items at the boundary of the diamonds are tightly bound, their clusters should not have been joined since each item has few connections to the opposite cluster and many slightly weaker connections to its own cluster. This failure to reconcile the degree of connectivity with the magnitude of a single connection is a shortcoming of distance-based approaches [3]. Similarly, the marginally larger separation between the singletons and their nearest neighbors, relative to other inter-item

distances, should have been offset by the many connections between them.

Complete Linkage and k-means favor convex clusters owing to their shared goal of minimizing intra-cluster variance. They fail to obtain a good partitioning of the Crescentric data set [4] in Figures 1(b) and 1(c) since its non-convex crescents have intra-cluster distances that occasionally exceed inter-cluster distances. In these two examples, it is evident that an item's cluster membership does not necessarily influence a nearby neighbor's membership. In a connectivity-based approach, items would instead exert an influence over one another, allowing clusters to become elongated or protruding.

The concept of connectivity may be introduced to clustering through graph partitioning, which represents each item as a node in a graph and assigns graph edge weights according to inter-item affinities. This analogy is the origin of spectral clustering methods, which analyze the eigenvalues and eigenvectors of a matrix describing graph connectivity. In principle, eigenvector components are a function of the entire graph. Therefore, eigenvectors have the potential to represent the global inter-item connectivity lacking in distance-based methods. By utilizing this global information, the spectral macrostate data clustering algorithm developed by Korenblum and Shalloway [5] outperforms k-means and the hierarchical clustering algorithms with respect to Two Diamonds (Figure 2(a)) and Crescentric (Figure 2(b)). Related spectral clustering methods have been shown to be superior to k-means across several synthetic benchmarks [6, 7] and have been applied across diverse areas including VLSI circuit partitioning [8, 9, 10, 11], image segmentation [12], load balancing [13, 14], and protein structure comparison [3].

Graph partitioning methods represent a graph in terms of a symmetric *affinity* or *adjacency matrix* A, whose entries are non-negative, inter-item affinities. An early approach to spectral graph partitioning [15] used the first k eigenvectors of A as its best low-rank approximation. These eigenvectors were projected onto a discrete, feasible solution with $w_\alpha(i) \in \{-1, 1\}$ by an optimization problem formulated as a constrained linear program.

Many popular approaches to graph bipartitioning are based on objective function minimization, which typically seeks to balance partition sizes and to minimize cut edges that cross partitions. Introducing the balanced partition constraint makes the problem NP-complete [16], thus motivating heuristic approaches. Fiedler recognized that the second smallest eigenvalue of the graph *Laplacian* is the optimal solution to a related, continuous problem in which the $\boldsymbol{w}_\alpha(i)$ are real valued [17]. This optimal solution is obtained by the

corresponding eigenvector–the *Fiedler vector*. The symmetric Laplacian [18] matrix from which it is derived has off-diagonal elements corresponding to the sign-inverted elements of A and diagonal elements that are sums over the corresponding rows of A:

$$L_{ij} = \delta_{ij} \left(\sum_k A_{ik} \right) - A_{ij} = \begin{cases} \sum_k A_{ik} & \text{if } i = j \\ -A_{ij} & \text{if } i \neq j \text{ and } i \text{ and } j \text{ are connected} \\ 0 & \text{otherwise .} \end{cases}$$

The optimality of the Fiedler vector spurred a number of approaches that relax the intractable, discrete problem to a continuous one based on spectral analysis of the (possibly normalized) Laplacian [12, 14, 19, 20]. The real-valued eigenvector components are then assigned to discrete partitions via *thresholding* according to eigencomponent sign, median value, or a large gap between adjacent sorted components. Hall [21] considered the related problem of optimally placing connected nodes in an r-dimensional space and found that each eigenvector of the Laplacian supplied one dimension of the coordinate. The Fiedler approach is readily extended to multiple clusters through recursive spectral bipartitioning [13]. However, direct partitional m-way clustering approaches [7, 9, 10, 22, 23, 24, 25, 26] have been shown to give better performance.

Real-valued cluster assignments implicitly express uncertainty. Unlike graph *partitioning*, which is often invoked from within a closed system that requires a hard assignment, to, for example, distribute computation or data, *data clustering* results are frequently subject to human analysis. The degree of certainty with which an item is assigned to a cluster therefore provides additional information during analysis. For example, low certainty may focus attention on those items requiring manual classification. The relatively greater assignment ambiguity of items with low certainties also indicates they are the items most likely to be classified differently by alternate clustering algorithms.

Objective functions adopted from graph partitioning, such as those that minimize edge cuts, have proven valuable in making relative comparisons: they have been used to search for an optimal partition when thresholding [11] and to assign items that are difficult to classify through heuristics [10]. They have also been used to derive theoretical bounds on partitioning quality [27, 28, 29, 30]. However, these objective functions do not support the intended use of data clustering for manual analysis, where the quality of a cluster is reflected in the degree to which it minimizes uncertainty. Imposing constraints on the real-valued assignment functions makes possible a probabilistic interpretation and an objective function

that quantifies cluster cohesion and isolation by measuring the degree of cluster fuzziness. A minimum uncertainty principle then prescribes how to define clusters and provides an absolute measure of clustering quality.

Korenblum and Shalloway [5] have previously described a data clustering method that (1) optimizes an objective function that measures the information content of the clustered representation, (2) automatically determines the optimal number of clusters m, and (3) provides fuzzy clustering assignment probability information. To accomplish this they used a stochastic model for data clustering based on the working hypothesis that each item selected for analysis has been statistically sampled from a continuous density distribution $p_B(\vec{x})$ of possibilities in a d-dimensional dataspace. This might reflect experimental selection from a continuous distribution of items (e.g., if members of a large population have been randomly selected for analysis) or might reflect complete analysis of a finite population that was naturally selected from a continuous distribution of possibilities (e.g., as in the case of a complete gene expression level analysis). If p_B is concentrated in separable subregions, then it is natural to dissect the possibility space along the corresponding subregion boundaries. The subregions are called *macrostates* and the items or *microstates* lying within each macrostate are gathered into a cluster. A similar dynamical metaphor motivates the growing body of diffusion-based clustering approaches [5, 31, 32, 33, 34, 35, 36] that consider random walks over weighted graphs.

The macrostate data clustering proposed by Korenblum and Shalloway is a fuzzy, spectral, m-way clustering approach in which the $\boldsymbol{w}_\alpha$ are a linear combination of the first m eigenvectors, chosen so that all values lie between zero and one to allow a probabilistic interpretation. Sec. II A shows how macrostates would be computed in the hypothetical case that p_B were known *a priori*. This approach is heuristically motivated by macrostate dissection methods that were previously developed for stochastic dynamic systems described by the Smoluchowski equation. Sec. II B describes show how this procedure can be adapted to data clustering when the underlying p_B is not known. As desired, this converts the clustering problem into an optimization problem using an information-dependent objective function and achieves the goals set above.

The resultant non-linear information optimization problem was previously solved by a brute-force technique that worked for modest size problems ($N = 200$), but which could not solve the larger problems [e.g., $N \sim O(10^4)$] that emerge in areas of interest such as

gene chip analysis [37]. This paper describes an efficient algorithm that scales to problem sizes of biological interest. Several other extensions accommodate biological data sets: the theory has been adapted in Sec. II D to handle asymmetric dissimilarity matrices, such as those derived from DNA sequence alignment scores [38] and the algorithm may be invoked recursively to expose hierarchical substructure, such as that arising during the classification of proteins [39].

A practical two-phased solution to the global optimization problem is described in Sec. III. An approximate solver determines a set of real-valued $\boldsymbol{w}_\alpha$, whose components may not precisely satisfy the probabilistic requirements. Nevertheless, thresholding the approximate solution is an alternative to other hard clustering methods that define m clusters from m eigenvectors. As the approximate solver builds clusters from m *representatives*, or items strongly identified with a particular cluster, it is particularly relevant to methods based on the similar ideas of prototypes [10] and sign structures [23]. In the limit in which the clusters are completely isolated or disconnected, the approximate solution is exact. In other cases, it is refined to yield a fuzzy clustering via the second phase, constrained solver. Sec. IV applies the solver to several large problems and Sec. V discusses its broader applicability to molecular dynamics simulations and the hierarchical classification of protein structures.

II. THEORY

A. Continuous Macrostate Dissection

Coarse graining in nonequilibrium statistical physics reduces the complexity of a complete, microscopic description of a dynamical system to a simpler model that captures its essential, macroscopic features [40]. This technique projects a configuration of rapidly fluctuating microscopic states (microstates) onto a set of more slowly varying macroscopic states (macrostates). For example, a classical, microscopic description of a protein is provided by the bond lengths, bond angles, and dihedral angles associated with atoms along the backbone. Such a system may be approximated by the dihedral angles alone owing to the high-frequency, small-scale bond length vibrations and the near rigidity of the bond angles. The resulting Ryckaert-Bellemans [41] potential of a single dihedral angle is shown in Figure 3. It remains characterized by the full range of the angular coordinate, though

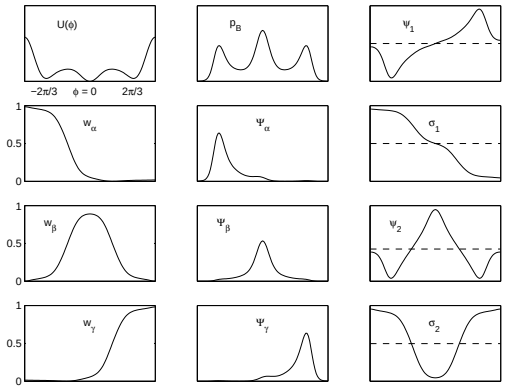


FIG. 3: Macrostate dissection of Ryckaert-Bellemans dihedral angle potential. The Ryckaert-Bellemans potential $U(\phi)$ gives rise to the equilibrium distribution $p_B \propto e^{-\beta U(\phi)}$. Given p_B, the Smoluchowski operator is defined according to Eq. (2) and discretized. Spectral analysis of the operator yields the eigenfunctions, of which $\psi_0 = p_B$, ψ_1, and ψ_2 are shown. Two non-trivial $\sigma_n = \psi_n/\psi_0$ are also shown. The nodal surfaces of the σ_n separate macrostates, as in the ψ_n. In addition, the structure of the σ_n is approximately level within a macrostate. Linear combinations of σ_0, σ_1, and σ_2 define the assignment functions w_α, w_β, and w_γ, which in turn filter out the macrostate regions from p_B to define macrostate distributions Ψ_α, Ψ_β, and Ψ_γ, respectively.

its well-defined catchment regions separate the Gibbs-Boltzmann equilibrium probability distribution, p_B, into three macrostates corresponding to the single trans and two gauche conformations. Consideration of dihedral angles across the entire protein may lead to further coarsening into "folded" and "unfolded" states. In such high-dimensional systems, visual determination of macrostates is not practical: a means of automatically dissecting configuration space into macrostate regions, such as the regions Ψ_α, Ψ_β, and Ψ_γ of the single dihedral angle shown in Figure 3, is required.

Dissection of the equilibrium probability distribution may be accomplished by coarse graining an initial system description provided by a fine-grained differential equation over microstates into a master equation whose relatively fewer degrees of freedom correspond to the system's metastable macrostates. In the case of highly interactive, overdamped systems, such as those arising from protein dynamics, an appropriate fine-grained model is the diffusion equation. As a further simplification, the momentum distribution may by

taken close to equilibrium, so that probability evolves independently of momentum according to the Smoluchowski equation [42]

$$\frac{\partial p}{\partial t} = \vec{\nabla} \cdot \left[p_B \, \vec{\nabla} \left(p_B^{-1} p \right) \right] , \tag{2}$$

where p_B is the potential- and temperature-dependent Gibbs-Boltzmann distribution. In general, the equation would account for anisotropic diffusion through an arbitrary symmetric tensor. However, the tensor may be brought to a diagonal form through orthogonal transformation of coordinates and to a multiple of the identity by rescaling the coordinates. The resulting scalar may then by absorbed into the time coordinate as done in Eq. (2). To facilitate eventual transition to a discrete system, Eq. (2) may be written in operator form as

$$\frac{\partial p(x,t)}{\partial t} = \int \Gamma(x,x') \, p(x',t) \, dx' \, . \tag{3}$$

Formally, the equation is solved via the spectral expansion

$$p(\vec{x},t) = \sum_{n=0}^{\infty} c_n \, e^{-\gamma_n \, t} \, \psi_n(\vec{x}) \, , \tag{4}$$

where the non-negative[1] eigenvalues γ_n and the right eigenfunctions ψ_n satisfy

$$\Gamma \, \psi_n = \gamma_n \, \psi_n \, . \tag{5}$$

From Eq. (2) it is evident that p_B is a stationary solution to the Smoluchowski equation

$$\Gamma \, p_B = 0 \, .$$

Therefore, p_B must be proportional to the ground-state eigenfunction ψ_0, with corresponding eigenvalue $\gamma = 0$. Without loss of generality, ψ_0 may be normalized such that $c_0 = 1$. Eq. (4) then describes the relaxation of an initial distribution $p(\vec{x},0)$ to its equilibrium state $\psi_0(\vec{x})$

$$\lim_{t \to \infty} p(\vec{x},t) = \psi_0 \, , \tag{6}$$

with relaxation rates given by the eigenvalues. Since the probability distribution equilibrates to the Gibbs-Boltzmann distribution,

$$\psi_0 = p_B \, ,$$

[1] The eigenfunctions corresponding to negative eigenvalues are not normalizable

so long as the equilibrium state is non-degenerate. A d-fold degenerate equilibrium state results when p_B has support in d disjoint regions. Such a situation is characterized by d zero eigenvalues, but is easily resolved in advance and so need not be considered except for some numerical issues discussed later.

As discussed in greater detail in Refs. 43 and 44, a concentration of p_B into m separable macrostate regions will be reflected in a gap in the relaxation rates

$$0 = \gamma_0 < \gamma_1 < ...\gamma_{m-1} \ll \gamma_m \ .$$

The gap partitions relaxation dynamics into two distinct phases: fast, localized probability waves bring each macrostate to its internal equilibrium, while the $m-1$ non-trivial low-frequency eigenfunctions redistribute probability across macrostates to reach a global equilibrium. Probability is transported according to the exponential decay of eigenfunction amplitudes and flows between regions whose amplitudes differ in sign. Hence, nodal surfaces in the low-lying eigenfunctions separate macrostates. Since the low-lying eigenfunctions primarily transport probability across, rather than within, macrostates, the integrated probabilities within each macrostate, the *macrostate occupation probabilities* $p_\alpha^{\rm occ}$, remain approximately constant during the slow relaxation phase. As the disparity between rates across the two phases increases, the macrostates become more distinctly separated from one another. Hence the *minimum gap parameter* ρ_γ

$$\gamma_n/\gamma_{n-1} \ < \ \rho_\gamma \qquad (1 < n < m) \tag{7a}$$

$$\gamma_m/\gamma_{m-1} \ > \ \rho_\gamma \ , \tag{7b}$$

determines the number of clusters and expresses a desired tolerance for the quality of those clusters.

Nodal surfaces have previously been used to determine macrostate boundaries. In the case $m = 2$, the sign of eigenvector amplitudes has been used to separate clusters in spectral bipartitioning. This technique was generalized to a larger number of "aggregates" by Deuflhard et al. [23], who projected the m amplitudes of a single ensemble member i onto an m-vector describing their sign

$$(\text{sign}(\psi_0)_i, \text{sign}(\psi_1)_i, \ldots, \text{sign}(\psi_{m-1})_i) \ .$$

The authors show that this sign structure uniquely assigns each member to a cluster. Unfortunately, the sign structure may be heavily perturbed so that ambiguities exist for near-zero

amplitudes. The authors arrive at a hard clustering that resolves the ambiguities, but does not preserve the uncertainty of the ambiguous ensemble members.

Macrostate dissection linearly relates the m low-frequency eigenfunctions to a set of m continuous, fuzzy macrostate *assignment functions* $\{w_\alpha(\vec{x}) : \alpha = 1, \ldots, m\}$

$$w_\alpha(\vec{x}) = \sum_{n=0}^{m-1} M_{\alpha n}\, \sigma_n(\vec{x}) \ , \tag{8}$$

where

$$\sigma_n(\vec{x}) \equiv \psi_n(\vec{x})/\psi_0(\vec{x}) \ . \tag{9}$$

The non-negativity of ψ_0 guarantees that the low-lying σ_n, like the low-lying ψ_n, have nodal surfaces at macrostate boundaries. Renormalizing by ψ_0 effectively smooths out intra-macrostate amplitude fluctuations so that the σ_n are approximately constant within a macrostate when $n < m$ [2]. Fig. 3 in this paper, Fig. 3 in Ref. 43, and Fig. 1 in Ref. 44 give pictorial examples of the relationship between ψ_n, σ_n, and w_α.

Separation along nodal surfaces allows selection of coefficients $M_{\alpha n}$ so that, far from macrostate boundaries, the assignment functions approximate a hard partitioning by assigning ensemble members to clusters with near certainty:

$$w_\alpha(\vec{x}) \approx 0 \text{ or } 1 \qquad \text{(away from macrostate boundaries)} \ . \tag{10}$$

Near the macrostate transition regions the assignment functions may take on intermediate values, so long as they satisfy

$$w_\alpha(\vec{x}) \ \geq 0 \ , \qquad \forall \alpha, \vec{x} \tag{11a}$$

$$\sum_\alpha w_\alpha(\vec{x}) \ \ = 1 \ , \qquad \forall \vec{x} \ . \tag{11b}$$

These conditions ensure that the assignment functions cover the entire space, so that they effectively act as filters to fractionally select those ensemble members that belong to the corresponding macrostate. Fig. 3 shows that the three assignment functions w_α, w_β, and w_γ isolate the three concentrated regions Ψ_α, Ψ_β, and Ψ_γ of the equilibrium distribution p_B.

The assignment functions express uncertainty whenever they deviate appreciably from zero or unity. Thus, while the most informative dissection of space is a strictly binary,

[2] Only the low-frequency σ_n will take constant values away from macrostate boundaries. The σ_n ($n \geq m$), like their corresponding ψ_n, will vary within the macrostate regions.

hard assignment, the least informative assigns each ensemble member to each macrostate with equal likelihood $1/m$. Intuitively, the degree of uncertainty is related to the amount of overlap between assignment functions w_α and $w_{\beta\neq\alpha}$. This may be quantified by the uncertainty

$$\Upsilon_\alpha(M) \equiv \frac{\sum_{\beta\neq\alpha} \int w_\alpha(\vec{x})\, w_\beta(\vec{x})\, p_B(\vec{x})\, d\vec{x}}{\int w_\alpha(\vec{x})\, p_B(\vec{x})\, d\vec{x}} \,, \tag{12}$$

which is an entropic measure in the information-theoretic sense in that it increases with increasing ignorance of the system. M is included as an argument to the uncertainty Υ_α since it is indirectly dependent on the $M_{\alpha n}$ through the assignment functions.

An alternate derivation of uncertainty proceeds from a filter analogy, in which the w_α are viewed as devices for preparing and observing statistical ensembles. Fig. 3 depicts the preparation of p_B in three alternate macrostates. For example, the distribution prepared in macrostate α at $t = 0$ is

$$\Psi_\alpha(\vec{x}, 0) = \frac{w_\alpha(\vec{x})\, p_B(\vec{x})}{\int w_\alpha\, p_B(\vec{x})\, d\vec{x}} \,, \tag{13}$$

where Ψ_α has been normalized to satisfy conservation of probability. The definition filters the equilibrium distribution with an assignment function to select the appropriate ensemble members.

Since the expectation value of the observable $\mathcal{O}(\vec{x})$ given probability distribution Ψ is

$$< \mathcal{O} >= \int \mathcal{O}(\vec{x})\, \Psi(\vec{x}, t)\, d\vec{x} \,,$$

the probability that an ensemble member in state Ψ_α is observed in macrostate α is defined as

$$p_\alpha^{\rm obs}(\Psi_\alpha; t) = \int w_\alpha(\vec{x})\, \Psi_\alpha(\vec{x}, t)\, d\vec{x} \,. \tag{14}$$

Because the w_α are constructed from the smooth low-frequency σ_n, they will be fuzzy and will have support in overlapping regions. Therefore, even a measurement on state $\Psi_\alpha(\vec{x}, 0)$ immediately following its preparation at $t = 0$ will have some probability of finding ensemble members in macrostates other than α: the probability $\overline{\Upsilon}_\alpha$ that an ensemble member prepared in macrostate α will be observed in that macrostate at $t = 0$ may be less than unity. Combining Eqs. (13) and (14) and relating the fraction to $\Upsilon_\alpha(M)$ through Eq. (12), yields

$$\overline{\Upsilon}_\alpha(M) \equiv p_\alpha^{\rm obs}(\Psi_\alpha; 0) = \frac{\int w_\alpha^2\, p_B\, d\vec{x}}{\int w_\alpha\, p_B\, d\vec{x}} = 1 - \Upsilon_\alpha(M) \,. \tag{15}$$

The $\overline{\Upsilon}_\alpha(M)$ measure the certainties of the macrostate assignments ($0 \leq \overline{\Upsilon}_\alpha(M) \leq 1$). Since a quality dissection should not unduly sacrifice the certainty of one macrostate to

favor another, a good definition of the macrostates is one that maximizes the product (i.e., the geometric mean) of their certainties. Imposing this *minimum uncertainty criterion* is equivalent to choosing the $M_{\alpha n}$ that minimize the objective function

$$\Phi(M) \equiv -\sum_{\alpha} \log \overline{\Upsilon}_{\alpha}(M) \ . \tag{16}$$

In combination with the constraints imposed by Eq. (11), its minimization determines the $M_{\alpha n}$ to complete the macrostate dissection.

B. Macrostate Data Clustering: Symmetric Γ

Physical coarse graining may be adapted to fuzzy data clustering by viewing the continuous microstates describing the fine-grained physical system as discrete data items and the macrostates as clusters. The analog to Eq. (2), in which the Smoluchowski operator acts on the continuous probability distribution $p(\vec{x}, t)$, is

$$\frac{d\boldsymbol{p}(t)}{dt} = \Gamma \cdot \boldsymbol{p}(t) \ , \tag{17}$$

in which the transition rate matrix Γ acts on the probability vector $\boldsymbol{p}(t)$ of individual item probabilities $p_i(t)$.

Eq. (17) is consistent with a (biased) random walk approach to data clustering [5, 24, 31, 32, 34, 35, 36, 45]. Under this interpretation, probability diffuses along edges in a weighted graph. As in the physical system, a data set amenable to clustering exhibits a large disparity between fast, local fluctuations that equilibrate probability within a cluster and less frequent transitions that cross cluster boundaries. Edge weights are defined as a function of inter-item distance and are related to the probability or rate of diffusion between items. For example, the probability of transitioning in a single step from item i to item j may be defined as the weight between the two items normalized by the sum of the weights between i and each of its neighbors [24, 32, 36, 45]. The stochastic matrix so defined is the transition probability matrix associated with a discrete-time Markov chain.

In principle, Γ could be defined by discretizing the Smoluchowski operator in Eq. (2). Doing so requires knowledge of the equilibrium distribution p_B, which, while frequently available for a physical system, is in general unknown for a data set. Deriving it from the data through functional density estimation would be computationally inefficient. Instead,

it is possible to posit a form of Γ consistent with the assumed kinetic model. Tishby and Slonim [35] have argued that transition probabilities should be exponential in inter-item distance, since this form is consistent with the typical assumption that distances are additive and that probabilities are multiplicative. Belkin and Niyogi [31] have shown that Eq. (17) induces a Gaussian (i.e., exponential in distance squared) form, while Nadler et al. [34] have found that a normalized Gaussian form approximates the backward Fokker-Planck equation of a related system.

The present derivation follows Korenblum and Shalloway [5], who used dimensionality to motivate the form of Γ. Since the left-hand side of Eq. (17) has units of inverse time and the diffusion constant (implicit on the right-hand side since it was taken to be unity) has units of distance squared divided by time, Γ must have units of inverse distance squared, independent of dimensionality. This form accounts for the attenuation in transition rate across long distances, but not for potential occlusion by intervening items that would prevent direct transition between two items. Such interception of probability may be modeled by an exponential cutoff scaled to the mean nearest-neighbor squared distance $\langle d_0^2 \rangle$:

$$\Gamma_{ij} \;=\; \frac{e^{-(D_{ij})^2/2\langle d_0^2\rangle}}{(D_{ij})^2} \;, \qquad i \neq j \;, \tag{18a}$$

$$\langle d_0^2 \rangle \;=\; N^{-1} \sum_{i=1}^{N} (D_{i<})^2 \;, \tag{18b}$$

where $D_{i<}$ is the smallest element in the i^{th} row of D. The negative diagonal elements of Γ are fixed as

$$\Gamma_{ii} = -\sum_{j \neq i} \Gamma_{ji} \tag{19}$$

by the conservation of probability and the subsequent requirement that

$$\mathbf{1} \cdot \Gamma = 0 \;. \tag{20}$$

where

$$\mathbf{1}_i = 1 \;, \qquad \forall i \;.$$

The positivity of the off-diagonal elements

$$\Gamma_{ij} \geq 0 \;, \qquad i \neq j \;, \tag{21}$$

is necessary to ensure that the $p_i(t)$ remain non-negative under temporal evolution.

The left and right eigenvectors of Γ are identical and intrinsically orthogonal because of the symmetry of Eq. (18). They may be normalized to ensure the orthonormalization condition

$$\langle \boldsymbol{\psi}_n | \boldsymbol{\psi}_m \rangle = \delta_{nm} \ , \tag{22}$$

where the inner product is expressed in bra-ket notation

$$\langle \boldsymbol{x} | \boldsymbol{y} \rangle \equiv N^{-1} \, \boldsymbol{x} \cdot \boldsymbol{y} \ . \tag{23}$$

Eq. (20) and the symmetry of Γ imply that

$$\Gamma \cdot \mathbf{1} = 0 \ , \tag{24}$$

so that $\boldsymbol{\psi}_0$ is proportional to $\mathbf{1}$. Eq. (23) fixes normalization to give

$$\boldsymbol{\psi}_0 = \mathbf{1} \ . \tag{25}$$

As in a continuous system, Γ may have multiple stationary eigenvectors. In this case Eq. (25) will not hold, though the set of degenerate groundstates will span the $\mathbf{1}$ vector. Appendix B discusses a straightforward computational procedure for ensuring the validity of Eq. (25) when Γ is symmetric.

The constancy of $\boldsymbol{\psi}_0$ allows the discrete form of Eq. (8) to be expressed directly in terms of the $\boldsymbol{\psi}_n$, rather than the $\boldsymbol{\sigma}_n$

$$\boldsymbol{w}_\alpha = \sum_{n=0}^{m-1} M_{\alpha n} \, \boldsymbol{\psi}_n \ . \tag{26}$$

The discrete analogs of the constraints of Eq. (11)

$$\boldsymbol{w}_{\alpha,i} \geq 0 \qquad \forall \alpha, i \tag{27}$$

$$\sum_\alpha \boldsymbol{w}_{\alpha,i} = 1 \qquad \forall i \ , \tag{28}$$

may be written in terms of M

$$\boldsymbol{w}_{\alpha,i} \ = \ \overrightarrow{M}_\alpha \circ \overrightarrow{\boldsymbol{\psi}}_i \geq 0 \qquad \forall \alpha, i \tag{29}$$

$$\sum_\alpha \overrightarrow{M}_\alpha \ = \ \hat{\varepsilon}_0 \ , \tag{30}$$

where

$$\hat{\varepsilon}_0 \ = \ \langle \mathbf{1} | \overrightarrow{\boldsymbol{\psi}} \rangle \ , \tag{31}$$

$$\overrightarrow{M}_\alpha \ = \ (M_{\alpha 0}, \, M_{\alpha 1}, \, \ldots, \, M_{\alpha (m-1)}) \ ,$$

and $\overrightarrow{\boldsymbol{\psi}}$ is the supervector having components $(\boldsymbol{\psi}_0,\ \boldsymbol{\psi}_1,\ \ldots,\ \boldsymbol{\psi}_{m-1})$. When Eq. (25) holds, the right-hand side of Eq. (31) simplifies to $\boldsymbol{e}_1$, the vector whose first component is one and all others are zero. Under such circumstances, Eq. (30) reduces to

$$\sum_{\alpha} M_{\alpha n} = \delta_{n0} \ .$$

Items are assumed to be sampled from an unknown, continuous probability distribution p_B, so that a uniform average over a large set of items approximates an equilibrium-weighted average in a continuous space

$$\int f(\vec{x})\, p_B(\vec{x})\, d\vec{x} = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{f}(\vec{x}_i) = \lim_{N\to\infty} \langle \mathbf{1} | \boldsymbol{f} \rangle \ . \tag{32}$$

This motivates the following definition of the equilibrium vector

$$\boldsymbol{p}^{\mathrm{eq}} = \mathbf{1} \ , \tag{33}$$

which plays a role analogous to p_B in a continuous system and, like p_B, is identical to $\boldsymbol{\psi}_0$ in a non-degenerate system. Therefore, the integrals over p_B in the certainty $\overline{\Upsilon}_\alpha(M)$ of a physical system are approximated by sums over $\boldsymbol{p}^{\mathrm{eq}}$ and the certainty of a cluster α is

$$\overline{\Upsilon}_\alpha(M) = \frac{\langle \boldsymbol{w}_\alpha | \boldsymbol{w}_\alpha \rangle}{\langle \mathbf{1} | \boldsymbol{w}_\alpha \rangle} \ . \tag{34}$$

The objective function of Eq. (16) is unchanged from the continuous system: its minimization through judicious choice of an $m \times m$ matrix M that respects Eqs. (29) and (30) provides a fuzzy dissection of the items into m clusters.

C. Spectral Structure of Transition Rate Matrix Γ

Understanding the relation between the transition rate matrix Γ as used in macrostate data clustering and the the transition probability matrix P of a discrete-time Markov chain illuminates the spectral structure of the former through the extensive perturbative analysis in the literature on the latter. Further, the explicit relationship between Γ and P links macrostate data clustering to other diffusion-inspired spectral clustering methods, which are typically described via a discrete-time Markov chain.

A discrete-time Markov chain describes transitions of a process X from state X_{i-1} at time t_{i-1} to state X_i at time t_i, where $\Delta t \equiv t_i - t_{i-1}$ is the fixed time between each pair

of temporally-contiguous transitions. The chain respects the Markov property that asserts that the probability of transition is dependent solely on the state from which the process transitioned, independent of earlier states. Formally,

$$\Pr(X_{n+1} = x | X_n = x_n, ..., X_1 = x_1, X_0 = x_0) = \Pr(X_{n+1} = x | X_n = x_n) \ .$$

A discrete-time Markov chain is specified in terms of the (left) stochastic transition probability matrix P, satisfying

$$\sum_i P_{ij} \ = \ 1 \ , \tag{35a}$$

$$P_{ij} \ \geq \ 0 \ , \qquad \forall i, j \ , \tag{35b}$$

where P_{ij} is the probability of transitioning from state j to state i in a single step:

$$P_{ij} = \Pr(X_{n+1} = i | X_n = j) \ .$$

Therefore,

$$\boldsymbol{p}(t + \Delta t) = P \cdot \boldsymbol{p}(t) \ . \tag{36}$$

The transition rate matrix Γ is the generator of a continuous-time Markov chain. Unlike a discrete-time Markov chain, which transitions at fixed time intervals, a continuous-time Markov chain transitions after spending a variable, but memoryless, state holding time in the state from which it is transitioning. Probability evolves in a continuous-time Markov chain according to Eq. (17). The first-order expansion of this equation

$$\begin{aligned} \boldsymbol{p}(t + \Delta t) \ &\approx \ \boldsymbol{p}(t) + \Delta t \, \Gamma \cdot \boldsymbol{p}(t) \\ &= \ (I + \Delta t \, \Gamma) \cdot \boldsymbol{p}(t) \\ &\equiv \ P \cdot \boldsymbol{p}(t) \end{aligned}$$

establishes the relationship between the transition rate matrix Γ of a continuous-time Markov chain and the transition probability matrix P of a discrete-time Markov chain

$$P(\Delta t) = I + \Delta t \, \Gamma \ . \tag{37}$$

Eq. (37) defines a valid transition probability matrix for the range of Δt satisfying Eq. (35). Assuming that Γ is a transition rate matrix satisfying Eqs. (19), (20), and (21), a valid

choice of Δt is $\min_i -1/\Gamma_{ii}$. Then, Eq. (37) establishes the eigenvalues and right eigenvectors of P from the eigenvalues γ_n and right eigenvectors $\boldsymbol{\psi}_n$ of Γ

$$\begin{aligned} P \cdot \boldsymbol{\psi}_n &= (I + \Delta t\, \Gamma) \cdot \boldsymbol{\psi}_n \\ &= (1 + \gamma_n\, \Delta t)\, \boldsymbol{\psi}_n \ . \end{aligned}$$

Hence, the transition rate matrix Γ and the transition probability matrix P have identical right eigenvectors $\boldsymbol{\psi}_n$ with eigenvalues γ_n and $1 + \gamma_n\, \Delta t$, respectively.

The realization of Eq. (10) is critically dependent on the constancy or level structure of an eigenvector across items sharing a cluster. On the basis of a dynamical metaphor, the previous sections argued that at long times intra-cluster probability fluctuations equilibrate so that the dominant flow of probability within the $\boldsymbol{\psi}_n$ (or σ_n) occurs between clusters and that intra-cluster variation across eigenvector magnitudes is small. The sign structure of an eigenvector then establishes a gradient of probability flow from negative to positive components. While intuitive, this dynamical perspective is neither necessary nor analytically informative. The validity of this and other spectral clustering approaches may be seen directly from matrix perturbation theory, which quantifies the intuition that small perturbations to the stochastic matrix of a Markov chain [7, 23, 46, 47] or to a general matrix [48] along with a large separation in its eigenspectrum lead to small perturbations in the space spanned by the low-lying eigenvectors. These perturbative results depend on the structure of Γ, rather than on its accuracy in modeling imposed system dynamics, thus justifying the macrostate data clustering approach independent of any dynamical considerations.

Perturbation theory proceeds from a block-diagonal matrix (or a matrix that may be brought to this form via permutation), where each of the m blocks B_i corresponds to an isolated subset. The system will have a left invariant subspace of dimension m spanned by the m degenerate left eigenvectors with eigenvalue $\gamma_n = 0$, for a transition rate matrix, or $\gamma_n = 1$, for a transition probability matrix. Similarly, the system will have a right invariant subspace of the same dimension, though there is no distinction between left and right subspaces if the matrix is hermitian. By definition of the transition matrices (i.e., Eqs. (20) or (35a)), $\mathbf{1}$ will be in the left invariant subspace, though a numerical eigensystem solver has freedom within the degenerate subspace to return any set of m eigenvectors spanning it and is not required to produce $\mathbf{1}$ as one of these eigenvectors. Figure 4(a) shows two sets of eigenvectors that span the triply-degenerate subspace of the Degenerate

Spiral data set. The data set consists of three crescents that are separated to the point of isolation. Figure 5(a) shows the block-diagonal structure of the resulting Γ matrix, which effectively suppresses any transitions between the clusters. The degeneracy of the system is evident in the eigenspectrum of Figure 4(a), where the first three eigenvalues are zero to numerical precision. The three eigenvectors shown to the left of the eigenspectrum span the degenerate subspace, but $\boldsymbol{\psi}_0 \neq \mathbf{1}$ in violation of Eq. (25). Since Γ is symmetric, the procedure of Appendix B may be used to enforce this condition. The two resulting, non-trivial eigenvectors shown to the right of the eigenspectrum, along with $\boldsymbol{\psi}_0 = 1$, provide an alternate basis for the degenerate subspace.

The eigenvectors in each of the two alternate bases of Figure 4(a) are constant away from cluster boundaries. This is characteristic of degenerate systems: the block-diagonal form of Γ and $\mathbf{1}\cdot\Gamma = 0$ mean that $\mathbf{1}\cdot B_i = 0$. Hence the m indicator vectors $\{\chi_\alpha : \alpha = 1, \ldots, m\}$, with $\chi_\alpha(x) = 1$ if item x is in subset α and zero otherwise, may be taken as an orthogonal basis of the left degenerate subspace [23]. Indeed, up to the sign of $\boldsymbol{\psi}_1$, this indicator basis was returned by the numerical solver in the leftmost pane of Figure 4(a). Since any vector in the degenerate subspace may be expanded in the indicator basis, the eigenvectors of any valid, alternate basis must also be level within a cluster. The significance of the level structure or piecewise constancy of eigenvectors with respect to spectral clustering has previously been discussed by Meila and Shi [24, 45]. This level structure is inherited by the crisply-defined assignment functions, which fulfill Eq. (10) exactly, as shown in the figure.

Stewart [48] has shown that the eigenvectors corresponding to a cluster of eigenvalues are sensitive to small perturbations in the matrix elements, though the subspace they span remains nearly invariant. In particular, the stability of an invariant subspace may be expressed in terms of a ratio ϵ, whose numerator grows with the magnitude of the perturbations E to block-diagonal form A and whose denominator grows with the separation between eigenvalues in the low- (i.e., near-degenerate) and high-ends of the spectrum[3]. A hypothesis condition involving the magnitude of the perturbations and the separation in the eigenspectrum establishes the theorem result, namely that vectors spanning the invariant subspace

[3] Sections 4.4 and 4.5 of Ref. [48] make the dependence on both the matrix element perturbation and the eigenspectrum separation explicit, whereas subsequent work related specifically to Markov chains [46] also requires a separation in the eigenspectrum, but is less direct about its role in perturbations to the eigensystem.

of A are perturbed by $O(\epsilon)$ to span the invariant subspace of $A + E$, while the eigenvalues of $A + E$ are perturbed by $O(||E||^2)$ from those of A. The condition is similar to the gap condition of Eq. (7), in which the ratio of eigenvalues ensures not only a large separation between eigenvalues but also that the separation is globally significant and reflects a perturbed block-diagonal form of Γ in which the magnitude of E is small. Considering irreducible transition probability matrices, Deuflhard et al. [23] find that the level structure of the eigenvectors is preserved to a first-order perturbation in the magnitude of E and the inverse of the separation between the corresponding eigenvalue and the Perron root $\lambda_0 = 1$.

The perturbative results state that the level structure of a degenerate system is largely preserved for small perturbations to Γ that introduce inter-cluster communication to lift the degeneracy. In this respect spectral analysis reflects the global properties of a data set: clusters, defined according to the intra- and inter-cluster communication between items, are clearly discernible in the low-lying eigenvectors. Local, purely distance-based clustering methods have no equivalent means of determining, not merely the similarity of two items, but the relative similarity of those items given the global relations they share with others in the data set.

The near-level structure of the eigenvectors under perturbation allows the expansion of the assignment functions to approximate Eq. (10). An example of the effects of perturbation on an eigensystem is given by the Spiral data set in Figure 4(b), in which the three crescents of the Degenerate Spiral data set are brought closer to one another. The relative proximity of the three crescents allows inter-cluster communication that weakens intra-cluster cohesiveness: the degeneracy in the eigenspectrum has been broken, the low-lying eigenvectors are only approximately level, and the assignment functions are fuzzy. Since the system is non-degenerate, $\boldsymbol{\psi}_0 = 1$ and is not shown in the figure. As desired, the strong separation in the eigenspectrum allows construction of assignment functions that exhibit only minor perturbations from the degenerate case. The expected perturbation of the assignment functions could, in principle, be determined prior to their construction through the formal results of Stewart [48] and the perturbative parameter ϵ.

Eventually, a system suffers such significant perturbation that the eigenspectrum gap is lost and with it all eigenvector structure. When the three crescents overlap one another, as in the Collapsed Spiral data set of Figure 4(c), the block structure of Γ in Figure 5(c) is seriously degraded. The eigenspectrum shows no discernible gap and the eigenvectors,

failing to indicate the individual crescent clusters, bear no resemblance to those of the two previous data sets. The method correctly infers from the eigenspectrum that the data set does not have any well-separated clusters.

D. Macrostate Data Clustering: Asymmetric Γ

Rather than being derived from measurements during the clustering process, as is the symmetric distance metric of Eq. (1), dissimilarities may be provided directly as raw data. For example, protein sequences can be clustered based on dissimilarities defined as sequence-sequence alignment scores, such as BLAST E-values [49]. Such biological measures of similarity are frequently asymmetric. ProClust [50] effectively normalizes distances by the length of the first sequence in a pair to discourage two sequences from having a strong transitive link through domains shared with an intermediate sequence, despite lack of any direct, mutual similarity. PSI-BLAST E-values [51], which measure the reliability of a sequence comparison between a query and a target, are also inherently asymmetric: the underlying algorithm derives a score matrix based on the query and iteratively refines it according to the multiple alignments resulting from its comparison against a database. Such data may be symmetrized by taking an average [52] or maximum [3] over the pair or, for binary measures indicating the presence or absence of inter-protein similarity, by replacing the two ambiguous, asymmetric results with a single value derived from a more computationally-expensive algorithm [53]. However, symmetrizing data may not be well motivated, particularly in situations for which asymmetry was intentionally introduced to meet a specific need, as in ProClust.

The perturbative results due to Stewart [48] apply to general matrices, so that the existence of a spectral gap indicates that the eigenvectors of an asymmetric Γ will exhibit the near-level structure conducive to macrostate data clustering. Since an asymmetric matrix may have complex eigenvalues, the obvious generalization of the spectral gap condition of Eq. (7) involves a ratio of *magnitudes* of eigenvalues. This approach is consistent with Stewart's theory, which accommodates complex eigenvalues through the use of norms and absolute values. In the context of the dynamical interpretation given by Eq. (4), the complex components of eigenvalues represent oscillation between macrostates. The requirement that there be a large disparity in the magnitude of 'slow' and 'fast' modes makes the intuitive as-

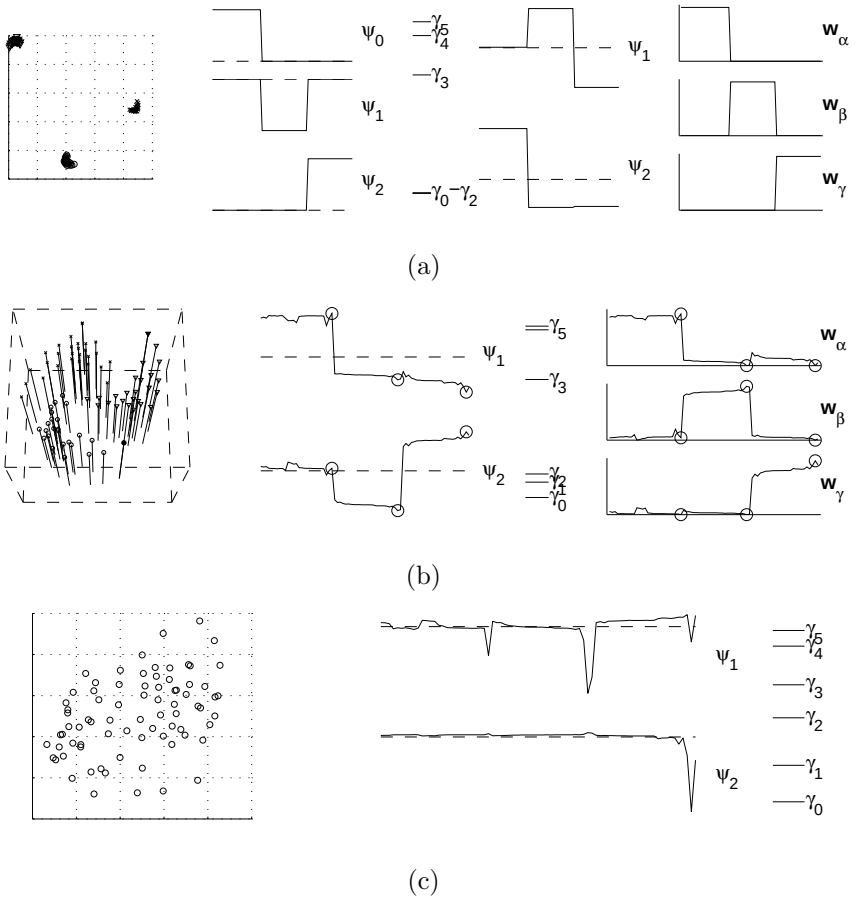


FIG. 4: Spectral analysis and macrostate data clustering of three crescents with varying degrees of separation. Each panel plots the data set, the eigenvectors $\boldsymbol{\psi}_n$, and the eigenvalues γ_n. Assignment functions $\boldsymbol{w}_\alpha$ are shown for those data sets admitting a clustering. (a) Degenerate Spiral. The extreme separation of the crescents leads to a triply degenerate system. The first column of three eigenvectors are one possible set that spans the degenerate subspace. To ensure that $\boldsymbol{\psi}_0$ is non-zero and adheres to Eq. (25), the procedure of Appendix B was applied to obtain $\boldsymbol{\psi}_0 = \mathbf{1}$ and the two subsequent eigenvectors pictured to the right of the eigenvalues. The eigenvectors, and hence the assignment functions, are level. (b) Spiral. The relative assignment strengths are displayed as the z coordinate overlaid on the two-dimensional plot of the data set. The nearness of the three crescents has broken the degeneracy, as reflected in the eigenspectrum. The eigenvectors are perturbed from a level structure, but the spectral gap allows for a valid clustering. (c) Collapsed Spiral. The three crescents have merged. No structure representative of the three crescents remains in the eigenvectors and no gap exists in the eigenspectrum: no macrostate data clustering is possible.

(a) (b) (c)

FIG. 5: Structure of Γ. Each element has been normalized by the average component of the Spiral Γ. Color gradient indicates the value of matrix elements, with $\Gamma_{ij} \leq 0$ shaded black and $\Gamma_{ij} \geq 1$ shaded white. The only negative values occur along the diagonal according to Eq. (21). (a) Degenerate Spiral. Block diagonal structure reflects the degeneracy of the system. (b) Spiral. The presence of off-diagonal elements breaks the degeneracy, but the matrix is an evident perturbation from a block-diagonal form. (c) Collapsed Spiral. The original block-diagonal structure is badly degraded.

sertion that the metastability of a macrostate depends not only on a slow rate of transition, but also on a slow rate of oscillation.

An asymmetric matrix gives rise to complex eigenvalues and eigenvectors. The left eigenvectors $\boldsymbol{\varphi}_n$ and the right eigenvectors $\boldsymbol{\psi}_n$ are uniquely defined and related through a bi-orthogonality condition

$$\langle \boldsymbol{\varphi}_n | \boldsymbol{\psi}_m \rangle = \delta_{nm} \ .$$

The definition of Γ continues to guarantee Eq. (20), with $\boldsymbol{\varphi}_0 = \mathbf{1}$. From the discussion in Sec. II C it follows that the m indicator vectors χ_α span the left degenerate subspace. Given that a groundstate of $\mathbf{1}$ simplifies the theory and that (minor perturbations to) the level structure of the indicator vectors are favorable to macrostate data clustering, it is natural to expand the assignment functions in the *left* eigenvectors.

In principle, the theory can directly accommodate expansion in the left eigenvectors. However, practical difficulties arise because of the imaginary components of the eigenvectors, their lack of internal orthogonality, and the potential invalidity of $\boldsymbol{\varphi}_0 = 1$ in a degenerate system. Since macrostate data clustering must take a linear combination of eigenvectors that

ensures the reality of the assignment functions, there is no loss of generality in constructing an equivalent *real* basis $\tilde{\boldsymbol{\varphi}}_n$ defined from linear combinations of the left eigenvectors. Complex eigenvalues and eigenvectors arise in complex conjugate pairs: if the j^{th} and $j{+}1^{st}$ eigenvalues form a complex conjugate pair with $\gamma_{j+1} = \gamma_j^*$, then the associated eigenvectors also form a complex conjugate pair with $\varphi_{j+1} = \varphi_j^*$. Therefore, the real basis may be defined as

$$\left.\begin{array}{rcll} \tilde{\boldsymbol{\varphi}}_n & = & \boldsymbol{\varphi}_n & \\ \tilde{\boldsymbol{\varphi}}_n & = & \frac{1}{2}\left(\boldsymbol{\varphi}_n + \boldsymbol{\varphi}_{n+1}\right) & \\ \tilde{\boldsymbol{\varphi}}_{n+1} & = & \frac{1}{2i}\left(\boldsymbol{\varphi}_n - \boldsymbol{\varphi}_{n+1}\right) & \end{array}\right\} \begin{array}{l} \text{if } \gamma_n \in \mathbb{R}\ , \\ \\ \text{if } \gamma_n \in \mathbb{C}\ . \end{array}$$

The lack of orthogonality of the left eigenvectors $\boldsymbol{\varphi}_n$, and the $\tilde{\boldsymbol{\varphi}}_n$, complicates the evaluation of the cluster certainty and the form of the equality constraints. Eqs. (25) and (26) yield a simplified form of the certainty of cluster α, as expressed in Eq. (34), when the $\boldsymbol{\psi}_n$ are orthonormal:

$$\overline{\Upsilon}_\alpha(M) = \frac{\sum_n M_{\alpha n}^2}{M_{\alpha 0}}\ .$$

Similarly, the brevity of the equality constraints of Eq. (30) relies on orthonormality to reduce the inner product of the $\boldsymbol{\psi}_n$ from Eq. (26) with the supervector $\overrightarrow{\boldsymbol{\psi}}$ to a Kronecker delta.

Since the $\tilde{\boldsymbol{\varphi}}_n$ are used in an expansion, it is the subspace they span that must be preserved rather than their individual forms. Therefore, it is possible and desirable to seek an alternate, orthogonal basis spanning the same subspace. If A is defined to be the matrix whose m columns are the $\tilde{\boldsymbol{\varphi}}_n$, then the singular value decomposition of A yields the desired orthogonal basis as the left singular vectors u_n. For a matrix $A \in \mathbb{R}^{N\times m}$, the (thin) singular value decomposition [54] is defined as

$$A = U\,\Sigma\,V^T\ ,$$

with the orthogonal matrices $U \in \mathbb{R}^{N\times m} = [u_0, \ldots, u_{m-1}]$ and $V \in \mathbb{R}^{m\times m} = [v_0, \ldots, v_{m-1}]$ of left singular vectors u_n and right singular vectors v_m and with the matrix $\Sigma \in \mathbb{R}^{m\times m}$ of singular values σ_n. The non-negative singular values are ordered

$$\sigma_0 \geq \ldots \geq \sigma_{r-1} > \sigma_r = \ldots = \sigma_{m-1} = 0$$

so that

$$\mathrm{rank}(A) = r\ .$$

Then, the null space of A is spanned by the $\{v_r, \ldots, v_{m-1}\}$ subset of the right singular vectors and the range of A is spanned by the $\{u_0, \ldots, u_{m-1}\}$ subset of the left singular vectors. Since the $\tilde{\boldsymbol{\varphi}}_n$ are linearly independent, A has full column rank $r = m$ and the desired orthogonal basis are the m columns of U.

Unfortunately, the singular value decomposition may not produce $\mathbf{1}$ as one of the left singular vectors. In fact, the $\tilde{\boldsymbol{\varphi}}_n$ from which the SVD is computed may not include $\mathbf{1}$ if the system is degenerate. Nevertheless, $\mathbf{1}$ is in the m-dimensional, left subspace spanned by the $\tilde{\boldsymbol{\varphi}}_n$. Therefore, projecting $\mathbf{1}$ out from the original left subspace reduces its dimensionality by one to $m-1$. Though the left subspace is accessible only through the m $\tilde{\boldsymbol{\varphi}}_n$, none of which are necessarily the null vector after the projection, SVD can extract the $m-1$ left singular vectors spanning the reduced space, which may then be augmented with $\mathbf{1}$ to construct the desired basis of m orthogonal vectors spanning the left subspace. Procedurally, the columns a_n of A are defined as

$$a_n \equiv \tilde{\boldsymbol{\varphi}}_n - \langle \tilde{\boldsymbol{\varphi}}_n | \mathbf{1} \rangle \, \mathbf{1} \; .$$

The SVD of A will have $m-1$ non-zero singular values. The difficulty of resolving "zero" given numerical inaccuracy may be avoided by selecting the $m-1$ left singular vectors corresponding to the largest singular values. The (complex) eigenvalues of the asymmetric Γ were used to determine m and to select the appropriate set of left eigenvectors for orthogonalization via SVD; the singular values from the SVD serve only to select out from this set of m vectors the $m-1$ spanning the reduced subspace that excludes $\mathbf{1}$. After being normalized to ensure the equivalent of Eq. (22), the orthogonal basis $\hat{\boldsymbol{\varphi}}$, comprised of elements from the set $\{\mathbf{1}, u_0, \ldots, u_{m-2}\}$, assumes the role of the eigenvectors $\boldsymbol{\psi}_n$ in Sec. II B. To avoid confusion with previous work, $\boldsymbol{\psi}_n$ are used when referring to symmetric Γ with the understanding that they should be replaced by $\hat{\boldsymbol{\varphi}}$ when Γ is asymmetric. This distinction is made explicit in the discussion of algorithmic control flow in Sec. III E.

III. COMPUTATIONAL METHODS

Defining macrostate data clusters requires minimizing the objective function $\Phi(M)$ subject to constraint Eqs. (29) and (30). Korenblum and Shalloway [5] solved this global, nonlinear optimization problem via a proof-of-concept, geometrical approach that iteratively

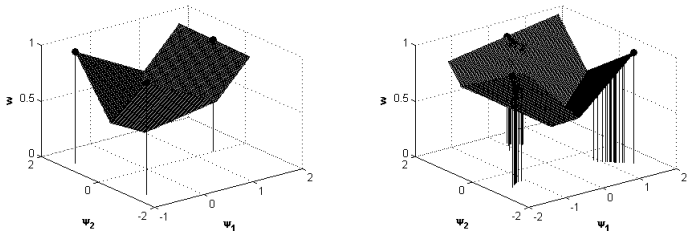
discovered constraints in order to enumerate candidate $\overrightarrow{M}_\alpha$. Their solution is reviewed briefly before the current two-phase, heuristic solution is discussed. The expected polynomial execution time of the two-phase solver make it considerably more efficient than the exhaustive enumeration of the previous brute-force, combinatoric approach.

$\Phi(M)$ is to be minimized as a function of the m^2 degrees of freedom of M within the feasible region defined by the $m \times N$ inequality constraints of Eq. (29) and the m equality constraints of Eq. (30). The equality constraints may be used to eliminate m degrees of freedom. Therefore, the inequality constraints are half-spaces that define a polytope as a feasible region within an $m(m-1)$-dimensional subspace. Korenblum and Shalloway have shown that a minimum of the constrained problem lies at a vertex of the polytope: a minimum is constrained by $m(m-1)$ "active" inequality constraints and all m equality constraints. The brute-force minimization routine initialized M to a fixed location within the polytope and chose a random direction along which to iteratively discover and travel along faces (of decreasing dimensionality) until a vertex was reached. This process of random enumeration continued until the same minimum was repeatedly discovered.

The first-phase of the current minimization routine, discussed in Sec. III A, solves m linear equations to find an approximate solution that satisfies the equality constraints but that may violate the inequality constraints, which are not explicitly considered. Sec. III B describes how this approximation may then be adjusted through a constrained linear program to satisfy the inequality constraints. The resulting solution is a fuzzy, m-way clustering of the input data set. Each of the m clusters may be recursively analyzed, as outlined in Sec. III C. Each invocation of the two-phased solver is preceeded by a routine that identifies outliers directly from analysis of the eigenvectors. The asymptotic complexity of each of the above modules as well as the specific libraries used in implementing them is presented in Sec. III E. In addition, a diagram makes explicit the flow of control between the modules.

A. Unconstrained Approximate Solution

The linear expansion of the assignment functions in Eq. (26) describes a hyperplane as a subspace of the $m+1$ dimensions $\boldsymbol{w}, \boldsymbol{\psi}_0, \boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_{m-1}$. Since the hyperplane is defined by the m coefficients $M_{\alpha 0}, M_{\alpha 1}, \ldots, M_{\alpha(m-1)}$ of the normal vector, any m equations describing



(a) (b)

FIG. 6: Assignment function hyperplanes. The intersections of the $\boldsymbol{w}_\alpha$, $\boldsymbol{w}_\beta$, and $\boldsymbol{w}_\gamma$ from the Degenerate Spiral and Spiral data sets of Figure 4. The largest assignment strength at a given coordinate is plotted to form the intersections of the hyperplanes. The impulses represent the items in the eigenspace representation. (a) Degenerate Spiral. The degeneracy collapses the items in the eigenspace representation to one of three coordinates. (b) Spiral. Lifting of the degeneracy perturbs the items from their eigenspace coordinates in (a).

(linearly independent) items in the plane suffice to fix the coefficients and uniquely characterize the hyperplane. Whenever $\boldsymbol{\psi}_0 = 1$, the hyperplane may be considered to be a function of the m variables $\boldsymbol{w}, \boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_{m-1}$. Such is the case in Figure 6, where the intersections of the assignment functions $\boldsymbol{w}_\alpha$, $\boldsymbol{w}_\beta$, and $\boldsymbol{w}_\gamma$ corresponding to the Degenerate Spiral and Spiral data sets from Figure 4 are plotted as a function of the non-constant low-lying eigenvectors $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$. At a given $(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2)$ coordinate, the largest assignment strength is plotted; this strength describes the highest probability with which an item is assigned to one of the m clusters.

The triply-degenerate zero eigenvalue of the Degenerate Spiral data set induces a perfectly level structure in the first three low-lying eigenvectors (see Figure 4(a)), which, in turn, collapses the N items in the eigenspace representation $\overrightarrow{\boldsymbol{\psi}}_i$ to three points. As expected, the nodal surfaces of the eigenvectors have segregated the items into three clusters according to their sign structure [23]: $(-,-)$, $(-,+)$, or $(+,0)$.

Macrostate data clustering is particularly straightforward for a degenerate system. Each cluster α may be associated with a representative item r_α, where the representatives are selected as the mutually most distance m items or simply as m items with mutually unique sign structures. Since each representative has the same eigenspace representation $\overrightarrow{\boldsymbol{\psi}}_{r_\alpha}$ as any other item i in the same cluster, the choice of a representative from amongst those items in a cluster is immaterial. The common representation of a representative and an item i

means that their assignment functions and constraints have the same expansion and are thus satisfied simultaneously. Therefore, a complete and optimal clustering over the N items is specified by a hard clustering that unambiguously assigns representative r_α to cluster α

$$w_\alpha(r_\beta) = \delta_{\alpha,\beta} \; . \tag{38}$$

Such an assignment clearly satisfies the equality constraints of Eqs. (27) and (28). Further, it is optimal since a hard clustering has no overlap between clusters.

Strictly degenerate systems are uncommon in practice. Fortunately, the above procedure may be generalized to non-degenerate data sets, where the fuzzy w_α are determined from M as opposed to being set as in Eq. (38). Therefore, it is instructive to solve the degenerate case by computing M rather than by assigning the w_α directly. From Eq. (26), Eq. (38) is written in terms of M as

$$\begin{pmatrix} (\boldsymbol{\psi}_0)_{r_\alpha} & (\boldsymbol{\psi}_1)_{r_\alpha} & \dots & (\boldsymbol{\psi}_{m-1})_{r_\alpha} \\ (\boldsymbol{\psi}_0)_{r_\beta} & (\boldsymbol{\psi}_1)_{r_\beta} & \dots & (\boldsymbol{\psi}_{m-1})_{r_\beta} \\ \dots & \dots & \dots & \dots \\ (\boldsymbol{\psi}_0)_{r_\mu} & (\boldsymbol{\psi}_1)_{r_\mu} & \dots & (\boldsymbol{\psi}_{m-1})_{r_\mu} \end{pmatrix} \cdot \begin{pmatrix} M_{\alpha 0} \\ M_{\alpha 1} \\ \vdots \\ M_{\alpha(m-1)} \end{pmatrix} = e_\alpha \; . \tag{39}$$

So long as the eigenspace representation of the representatives are linearly independent, i.e., the matrix on the left-hand side is non-singular, the above linear system may be solved to determine $\overrightarrow{M}_\alpha$. Hence, the solution of m such systems completely determines the assignment functions.

In a non-degenerate data set, the low-lying eigenvectors are perturbed from a piecewise constant structure (see Figure 4(b)) so that each item has a unique eigenspace representation, as shown in Figure 6(b). From this plot it is evident that the extremal items, i.e., those furthest removed from the origin, are the constrained items. Those items that are most strongly assigned to a cluster prevent its associated hyperplane from being shifted in the positive $\boldsymbol{w}$ direction, since doing so would force the assignment strength of one of these items to exceed unity. Conversely, those items that are weakly assigned to a cluster prevent the hyperplane from being shifted in the negative $\boldsymbol{w}$ direction, which might force their probability of assignment to become negative. When the constraints are satisfied for these extremal items, the linearity of the hyperplanes ensures that the constraints will also be satisfied for the intermediate items.

The assignment probabilities imposed on the representatives guarantee that the equality constraints are satisfied, not only for the representatives, but for all items. This may be seen by recognizing that the sum of m assignment function hyperplanes, of the form of Eq. (26), retains that form and is itself a hyperplane. The summation hyperplane is a subspace of the $m+1$ dimensions $\sum_\alpha \boldsymbol{w}_\alpha, \boldsymbol{\psi}_0, \boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_{m-1}$, with the m coefficients $\sum_\alpha M_{\alpha 0}, \sum_\alpha M_{\alpha 1}, \ldots, \sum_\alpha M_{\alpha(m-1)}$ of the normal vector. As such, m items suffice to define it. In particular, the m representatives r_α, all of which satisfy $\sum_\beta \boldsymbol{w}_\beta(r_\alpha) = 1$, uniquely define the hyperplane. The linearity of the hyperplane then implies that $\sum_\beta \boldsymbol{w}_\beta(i) = 1$ for all items i in the plane.

In an $m = 2$ dimensional system, choosing as the representatives the two most separated items and applying the above procedure would achieve the global minimum for $\Phi(M)$. The situation is complicated by increased dimensionality, where extremal representatives provide an approximate solution that may violate some inequality constraints, i.e., yield assignment probabilities less than zero or greater than one. For example, choosing extremal items as representatives for the Spiral data set forces a non-extremal item to have a negative assignment probability. Constraint violations, if they occur, will occur in the neighborhood of extremal items. Therefore, despite possible violations of the inequality constraints, the M defined by this approximate solver is an approximation to a valid, nearby solution. During the second phase of the minimization, the constrained solver explores the neighborhood of this approximation to determine a solution that does satisfy all constraints.

Korenblum and Shalloway have shown that a minimum arises when $m(m-1)$ inequality constraints are active. These constraints correspond to those items whose assignments are forced by a valid solution procedure to zero or unity, as done for the representatives. While the approximate solver does indeed make $m(m-1)$ inequality constraints active, they are restricted to involve one of the m representatives. In principle, the $m(m-1)$ active constraints need not involve a fixed set of m items. For example, the active inequality constraints for the $m = 3$ problem of Figure 6(b) involve four items. An obvious generalization to the current representative selection strategy is to choose, for each cluster under consideration, that item that is furthest from the origin and then to choose $m-1$ items that are furthest from it and that have unique sign structures. However, this generalization does not salvage the Spiral problem since it would continue to select the same $m = 3$ representatives as the more constrained approach. Therefore, as the approximate solver is already a heuristic approach,

little seems lost in restricting it to a single representative for each of the m clusters.

Representatives are selected by first using an N^2 comparison to choose those two items that are furthest separated in eigenspace. An iterative procedure than examines the remaining $O(N)$ items and extends the representative set to include that item whose minimal distance to any of the current representatives is maximal and whose sign structure differs from that of any current representative. Appendix C describes how the sign structure as used for representative selection differ from those proposed by Deuflhard et al. [23].

Viewed in the eigenspace representation $\overrightarrow{\psi}_i$, the representatives associate a direction (and magnitude) with each cluster. The separation between clusters along nodal eigenvector surfaces ensures that the representatives have a strong angular separation, while the relatively small perturbations from a level eigenvector structure group the items in a cluster near the associated eigenvector. Hence, Scott and Longuet-Higgins [22] noticed that the similarity of two items was strongest when the cosine of the angular between them approached unity. Similarly, Chan et al. [10] select an initial set of m prototypes according to magnitude and near orthogonality to all previously selected prototypes. They then assign any item whose eigenspace representation is within a small angle of the prototype to the associated cluster. Those items that are not within the tight angular cone of some prototype are combinatorially assigned to a cluster through use of a min-cut objective function. The process is iterated with the prototypes in subsequent rounds defined as the vector average of all items assigned to the corresponding cluster. Alpert et al. [9] describe a vector partitioning approach in which the eigenspace representation of items within a cluster are summed to form a vector. Maximizing an objective function that sums the squares of these vectors then partitions items according to both direction and magnitude.

Rather than considering a continuous direction, the magnitudes of item in the eigenspace representation may be projected onto their signs. This technique has frequently been applied to the Fiedler vector to threshold the continuous solution it describes onto a binary partitioning required by the problem statement [12, 14, 19, 20]. Deuflhard et al. [23] proposed sign structures as a higher-dimensionality generalization to thresholding or clustering according to sign. In their work, the sign structure of each item uniquely assigns it to a cluster, assuming the sign structure is stable with respect to perturbations. The sign structure of an item i is unstable if the magnitudes of all components of $\overrightarrow{\psi}_i$ are beneath a threshold, in which case the sign structure may not reflect that of the item's cluster, but

rather may be influenced by numerical noise around zero. To overcome this ambiguity, the authors expand a cluster characteristic function in the low-lying eigenvectors using a least squares approximation involving only those components corresponding to stable items. The coefficients of this expansion may then be used to determine approximate cluster characteristic functions over all items. An ambiguous item is then assigned to the cluster whose approximate characteristic function has the strongest signal for that item.

The procedure suggested by Deuflhard et al. is similar to that used by the approximate solver: both use the components of the low-lying m eigenvectors from a subset of the items in the expansion of a function whose i^{th} entry indicates the strength with which item i is assigned to the corresponding cluster. The approximate solver uses m representatives to establish the coefficients $M_{\alpha n}$ of the assignment functions, whereas Deuflhard et al. use a set of stable sign structures, with cardinality at least m, to determine the coefficients of cluster characteristic functions. The fundamental difference between the two approaches is that macrostate data clustering creates fuzzy assignment functions. Therefore, the solution provided by the approximate solver must be amended to satisfy the probabilistic interpretation provided by the constraints.

B. Constrained Solution from Approximation

Though it violates constraints, the solution μ found by the approximate solver is favorable with respect to the objective function $\Phi(\mu)$. The conditions upon which spectral clustering methods depend, the presence of a large gap in the eigenspectrum and small perturbations from a block-diagonal Γ, guarantee that perturbations to the eigenvectors $\boldsymbol{\psi}_n$ of Γ are also small. The constraints, M, and $\Phi(M)$, all of which are coupled to the $\boldsymbol{\psi}_n$, will also be mildly perturbed to the same order. Therefore, μ is the exact solution to an unperturbed system and is within a small perturbation of the actual solution to the perturbed system.

$\overrightarrow{\mu}$ may be considered to be a direction in the m^2-dimensional space that defines a solution, where it linearizes the rows of μ as $(\mu_{00}, \mu_{01}, \ldots, \mu_{0(m-1)} \ldots \mu_{(m-1)(m-1)})$. Seeking a nearby solution then amounts to picking a vector $\overrightarrow{M}$, defined analogously to $\overrightarrow{\mu}$, whose magnitude and direction are similar to $\overrightarrow{\mu}$. Thus, the non-linear objective function of Eq. (16) may be replaced by a linear objective function that maximizes the distance of $\overrightarrow{M}$ along $\overrightarrow{\mu}$, reducing the global optimization problem over the vertices of the polytope to a search localized in the direction of $\overrightarrow{\mu}$. Introducing the constraints allows the new optimization problem to be

$\overrightarrow{M}_\alpha$	-10^{-10}	-10^{-10}	1
$\overrightarrow{M}_\beta$	0.957988	-10^{-10}	-10^{-10}
$\overrightarrow{M}_\gamma$	0.042003	1	-10^{-10}

FIG. 7: M computed for Spiral data set as the solution of a linear program. Its components are small perturbations from the strictly binary form of μ found by the approximate solver.

formalized as the following linear programming problem

$$\begin{aligned} &\text{maximize} \quad \overrightarrow{\mu}^T \overrightarrow{M} \\ &\text{subject to} \;\; C \overrightarrow{M} \leq y \end{aligned}$$

where the C and y are defined to express the constraints of Eqs. (29) and (30).

The simplex method is a standard approach to solving linear programs. Though it has worst-case exponential running time, it has been shown to be efficient in practice: Spielman and Teng [55] analyzed its expected performance under Gaussian perturbation to arbitrary inputs, finding it to be polynomial in the dimensions of C and the standard deviation of the Gaussian perturbation. Applied to the Spiral data set, a simplex solver mildly perturbs the strictly binary form of μ to the M shown in Figure 7. The negative values are artifacts of the tolerance used in enforcing the constraints.

In principle, the first invocation of the linear programming solver simply returns M to the feasible region, where the inequality constraints are satisfied. It does not necessarily place M at a vertex (or even on a face) of the polytope, where the optimal solution lies. A second invocation of the solver, with a maximization condition that attempts to extend the solution vector along $\overrightarrow{M}$ instead of along $\overrightarrow{\mu}$, would ensure that the solution reaches the surface of the polytope. In practice, multiple iterations have not been necessary to achieve high-quality solutions.

C. Recursive Macrostate Data Clustering

Macrostate data clustering is partitional: unlike hierarchical methods, such as agglomerative clustering, which iteratively merge clusters, it may directly break a data set into the best defined $m \geq 2$ clusters. Even a nominally partitional strategy such as recursive bisection, which is constrained to solve $m = 2$ problems at each step, has a hierarchical

nature: a data set such as Crescentric Mosaic in Figure 8 would require two recursive bisections to discover the three pairs of crescents, representing the highest level structure, and then three additional bisections to differentiate the two crescents within a pair. A strictly partitional approach would treat Crescentric Mosaic as an $m = 6$ problem, thus looking past the high-level structure to focus directly on the constituent crescents.

Each of these approaches has the same drawback—an inability to directly convey to a researcher the different spatial scales of the data set and the clusters at each scale. In principle, agglomerative clustering should discover the three pairs and the six crescents. However, these will be hidden within the dendrogram created by the method to represent the data set's hierarchical structure. Recursive bisection will also discover clusters at different spatial scales. Unfortunately, it will also find spurious partitionings, such as a single pair of crescents separated from the other two pairs, which are necessary as transient steps to determine meaningful clusterings, such as the $m = 3$ partitioning of the three pairs, but which do not themselves consistently separate items according to spatial scale.

A strictly partitional approach may be efficient at immediately uncovering the most fine-grained structure of a data set, but it does so by sacrificing the high-level organization that will be of value to the researcher in assigning coarser relations between the items. Further, it effectively requires the ability to determine a local, per-item scale factor to replace $\langle d_0^2 \rangle$ in the formulation of Γ. Using the global scale factor $\langle d_0^2 \rangle$ defined in Eq. (18b) prevents macrostate data clustering from directly discovering the fine structure that separates crescents within a pair. Zelnik-Manor and Perona [26] have described a local scale based on the extent of an item's $k = 7$ nearest neighbors, which is applied to Crescentric Mosaic in Figure 8(a). In each pair of crescents, the two items which extend from one cluster nearest the other are mis-classified. Better results were obtained using $k = 2$ nearest neighbors, which shows the sensitivity of the approach to proper and problem-specific choice of k.

An intuitive recursive application of macrostate data clustering effectively performs partitional clustering at each of the data set's spatial scales. Recursive macrostate data clustering generalizes recursive bisection to allow arbitrary m-way fuzzy partitioning. After analyzing Γ at step s of the recursion to create the assignment functions $\boldsymbol{w}_\alpha^s$, any item i that is assigned with a threshold intensity to a cluster α through $\boldsymbol{w}_\alpha^s(i)$ is included in a transition matrix Γ_α. Analysis then proceeds recursively on Γ_α to discover any potential sub-structure in cluster α, which is described by assignment functions $\boldsymbol{w}_\beta^{s+1}(i)$. Two probabilities of assignment of

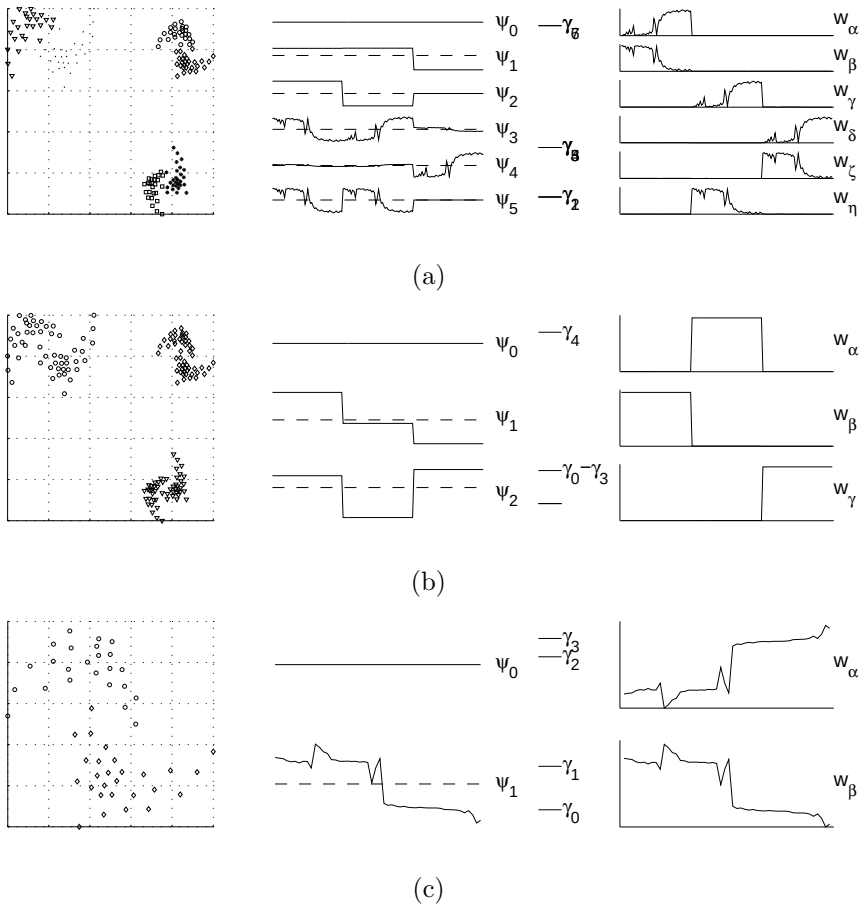


FIG. 8: Crescentric Mosaic data set, consisting of three pairs of crescents. (a) Data set clustered using the local, nearest-neighbor scale factor proposed by Zelnik-Manor and Perona [26]. The induced $m = 6$ clustering has several misclassifications near the boundaries of the crescents and fails to capture the highest-level structure of the data set. (b) First step of recursive macrostate data clustering discovers the data set's coarsest structure as an $m = 3$ problem. (c) One of the three $m = 2$ sub-problems that examines the fine-grained structure within a pair to differentiate the crescents. This case is representative of the other two sub-problems.

item i to cluster β at recursive step $s + 1$ are of interest: the conditional probability, given that item i has been assigned to cluster α at step s, is $\boldsymbol{w}_\beta^{s+1}(i)$, while the unconditional probability is $\boldsymbol{w}_\alpha^s(i) * \boldsymbol{w}_\beta^{s+1}(i)$. The choice of conditional or unconditional probability affects the certainties. The current implementation uses a threshold intensity of 0.5.

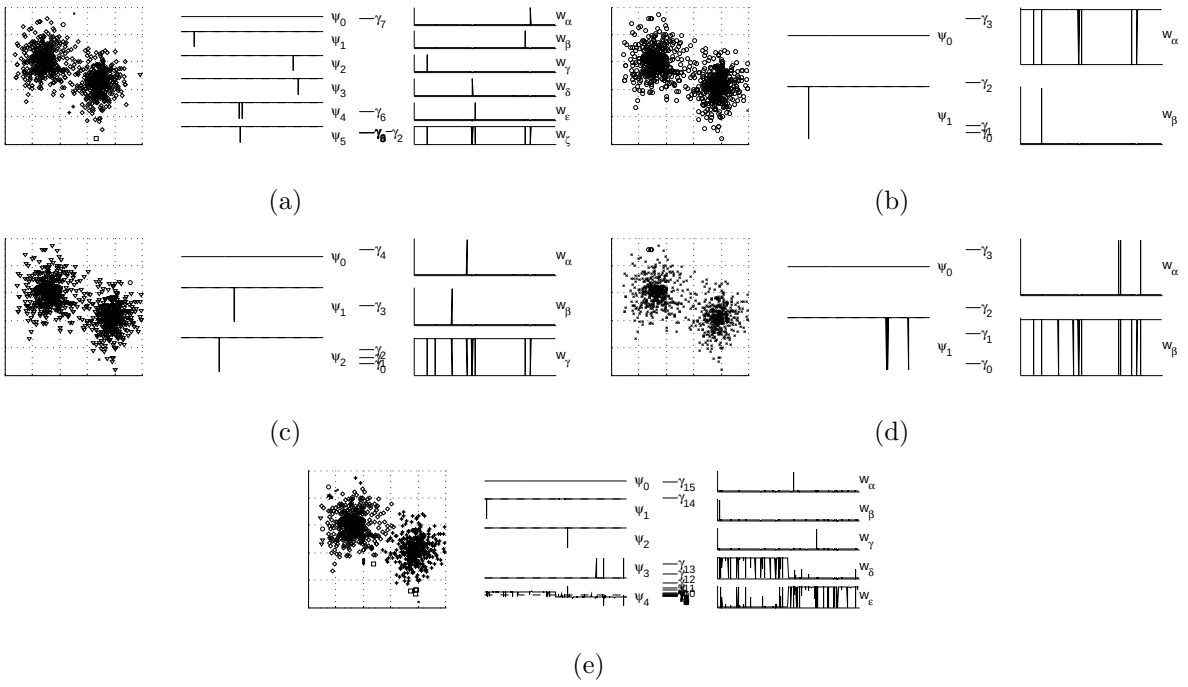


FIG. 9: Clouds data set clustered by treating outliers as isolated clusters. Outliers are effectively removed by the first four iterations of recursive clustering (a) - (d), until the eigenvector $\boldsymbol{\psi}_4$ of iteration five separates the two clouds. The final step solves an $m = 14$ problem, as reflected in the eigenspectrum; for space considerations, only five eigenvectors and assignment functions are displayed.

D. Outlier Detection

Data sets derived from experimentation are likely to contain noise, manifest as outliers—well-isolated clusters with few items. Outliers pose problems for data analysis because they may obfuscate more significant clusters. As described, macrostate data clustering copes with outliers by treating them as ordinary clusters: they are detected and effectively removed from the data set during recursion.

Figure 9 shows how a noisy data set consisting of two clouds whose items are sampled from a Gaussian distribution is handled by recursive macrostate data clustering. Four iterations of clustering remove a total of 23 outlying items. The two clouds are differentiated by eigenvector $\boldsymbol{\psi}_4$ in the fifth iteration during the solution to an $m = 14$ problem. For space considerations, Figure 9(e) presents only five of the 14 eigenvectors and resulting assignment functions.

This approach suffers several shortcomings. Though the outliers could be discarded by a post-processing step, the choice of $m = 14$ in the final step does not reflect the true bi-modal structure of the data set. Formulating the linear program of the optimization step as an $m = 14$ problem, rather than an $m = 2$ problem, introduces additional constraints, which negatively impact the spatial and temporal overhead of the constrained solver. Further, each iteration of outlier removal requires a costly recomputation of the eigensystem. Figure 9 shows that the removal of a few outliers often requires several iterations. Though the removal of a small number of well-separated items intuitively has little impact on the remaining data set, these items have a strong influence on $\langle d_0^2 \rangle$ in Eq. (18) since the distance $D_{i<}$ to their nearest neighbor is large. This results in a global affect on Γ_{ij} felt by all remaining items. An alternative scale factor to $\langle d_0^2 \rangle$, such as one defined on a per-item basis according to that item's local neighborhood [26], avoids this problem at the expense of introducing instability, as discussed in Sec. III C.

These concerns motivate direct removal of the outliers during a pre-processing step. Insofar as an outlier is an isolated cluster, it will be represented by an eigenvector with an impulse-like structure: the eigenvector will have nearly constant magnitude for the few items within the outlier and a different nearly constant magnitude for the many items external to it. For outliers appearing within the degenerate subspace, this idealized outlier signature may be present only as a linear combination of the degenerate eigenvectors. The determination of the boundary between the degenerate and non-degenerate subspaces is discussed in Appendix A.

An outlier signature is characterized by the total number of items N in the data set, the number of items N_{outlier} in the outlier, the signed magnitude $m_{\text{background}}$ of those items external to the outlier, and the signed magnitude m_{outlier} of the outlying items. In the non-degenerate subspace, an eigenvector $\boldsymbol{\psi}_{\text{outlier}}$ conforming to the outlier signature must be orthogonal to $\mathbf{1}$ according to Eq. (22). Therefore,

$$(N - N_{\text{outlier}})\, m_{\text{background}} + N_{\text{outlier}}\, m_{\text{outlier}} = 0 \ ,$$

and, assuming $N_{\text{outlier}} \ll N$, $m_{\text{background}} \approx -N_{\text{outlier}}\, m_{\text{outlier}}/N$. $m_{\text{background}}$ may then be related to m_{outlier} through the orthonormality of $\boldsymbol{\psi}_{\text{outlier}}$ and Eq. (23) to yield $m_{\text{outlier}} \approx \sqrt{N/N_{\text{outlier}}}$. Thus, an intuitive strategy examines a non-degenerate eigenvector to determine the least set of N_{outlier} items whose squared magnitudes nearly sum to N. The ambiguity

of this summation condition is removed by requiring that the squared magnitudes total a fraction f_{outlier} of N. Of course, the orthonormality condition ensures that the squared magnitudes of all N items sum to N. Therefore, a maximum size $N_{\text{outlier}}^{\max}$ is imposed on an outlier. The present implementation fixes $N_{\text{outlier}}^{\max} = 0.1 * N$ and $f_{\text{outlier}} = 0.95$.

Search for outliers is restricted to that subset of the non-degenerate eigenvectors whose eigenvalues γ_n precede the first eigenspectrum gap, i.e., where $n < m$. An outlier detected in the low-end of the spectrum is separated from other clusters by low-frequency transitions. However, the same signature applied to the high-end of the spectrum detects small clusters of items bound so closely to one another that interaction with other items is comparatively weak, even if the the distances involved are small on a global scale. When incorrectly applied to high-frequency eigenvectors, this technique has discovered 'outliers' within the dense core of a cloud of Gaussian items.

In contrast to $\boldsymbol{\psi}_{\text{outlier}}$ in the non-degenerate subspace, a vector $\boldsymbol{\psi}_{\text{outlier}}^{\text{degenerate}}$ representing an outlier signature in the degenerate subspace is not necessarily orthogonal to $\mathbf{1}$ and may not be identical to one of the eigenvectors, but rather may be constructed from a linear combination of eigenvectors

$$\boldsymbol{\psi}_{\text{outlier}}^{\text{degenerate}} = \sum_n c_n \boldsymbol{\psi}_n \ .$$

The vector may be re-scaled without destroying the impulse-like structure to ensure

$$\langle \boldsymbol{\psi}_{\text{outlier}}^{\text{degenerate}} | \boldsymbol{\psi}_{\text{outlier}}^{\text{degenerate}} \rangle = 1 \ . \tag{40}$$

Then, the orthonormality of the $\boldsymbol{\psi}_n$ constrains the summation over the expansion coefficients according to

$$\sum_n c_n^2 = N \ . \tag{41}$$

Since $\boldsymbol{\psi}_{\text{outlier}}^{\text{degenerate}}$ is not restricted to be orthogonal to $\mathbf{1}$, the magnitudes of outlying items need not be balanced by magnitudes of opposite sign corresponding to items external to the outlier. Therefore, a perfectly idealized outlier signature with $m_{\text{background}} = 0$ and $m_{\text{outlier}} = \sqrt{N/N_{\text{outlier}}}$ may be realized, in principle. If item i is a member of an outlier hidden within the degenerate subspace, the outlier signature may be extracted by maximizing $\boldsymbol{\psi}_{\text{outlier}}^{\text{degenerate}}(i)$ over the variables c_n defining it, according to

$$\frac{\partial}{\partial c_n} \left[\boldsymbol{\psi}_{\text{outlier}}^{\text{degenerate}}(i) - \frac{\lambda}{2} \left(\sum_{n'} c_{n'}^2 - N \right) \right] = 0 \qquad \forall n \ ,$$

where the Lagrange multiplier λ fixes Eq. (41). Therefore, the idealized outlier signature, if one exists for item i, may be constructed by taking $c_n = \boldsymbol{\psi}_n(i)/\lambda$, where $\lambda = \sqrt{\sum_n \boldsymbol{\psi}_n(i)^2}$ is determined from Eq. (40). The entire procedure for determining outliers within the degenerate subspace is then to construct the linear combination $\boldsymbol{\psi}_{\text{outlier}}^{\text{degenerate}}$ for each i and to assign to an outlier any items j that satisfy the conditions described above with respect to a non-degenerate eigenvector.

Outliers appear within both the degenerate and the non-degenerate subsets in the Clouds data set. The data set is subjected to five iterations of outlier removal in Figure 10 before an $m = 2$ problem involving the two clouds emerges in the final iteration. Direct outlier removal was thus successful in reflecting the subjective bi-clustering and in reducing the complexity of the problem posed as a linear program. In each of the first five iterations, outliers are removed and the remaining data are recursively analyzed. Outliers are detected within the non-degenerate subspace in iterations (c) and (e) and within the degenerate subspace in iterations (a), (b), and (d). For these latter cases, the constructed linear combinations $\boldsymbol{\psi}_{\text{outlier}}^{\text{degenerate}}$ are displayed. As expected, an identical linear combination results for items belonging to the same outlier.

Direct treatment of outliers did not reduce the number of invocations of the eigensystem solver on nearly identical data sets. Evidently, the outlier search is too conservative in discovering outliers when prevented from crossing the first eigenspectrum gap. This policy may be relaxed such that outliers are detected within the first $f_{\text{range}} * N$ eigenvectors, with $f_{\text{range}} = 0.1$, for example. Unfortunately, this may lead to excessive recursion since the likelihood of detecting an outlier increases as more of the spectrum is searched and since outlier detection leads to re-analysis of the data set without an attempt to find non-outlier clusters. Hence, outlier detection is permitted to extend beyond the first gap, so long as all but two low-lying eigenvectors either exhibit an outlier signature or may be used to construct a linear combination that does so. If two or more low-lying eigenvectors are uninvolved in outlier detection, they may be used to define non-outlier clusters; no outliers are removed and clustering proceeds according to Eq. (26). Figure 11 shows that the procedure reduces the number of invocations of the eigensystem solver from five (see Figure 9) or six (see Figure 10) to three.

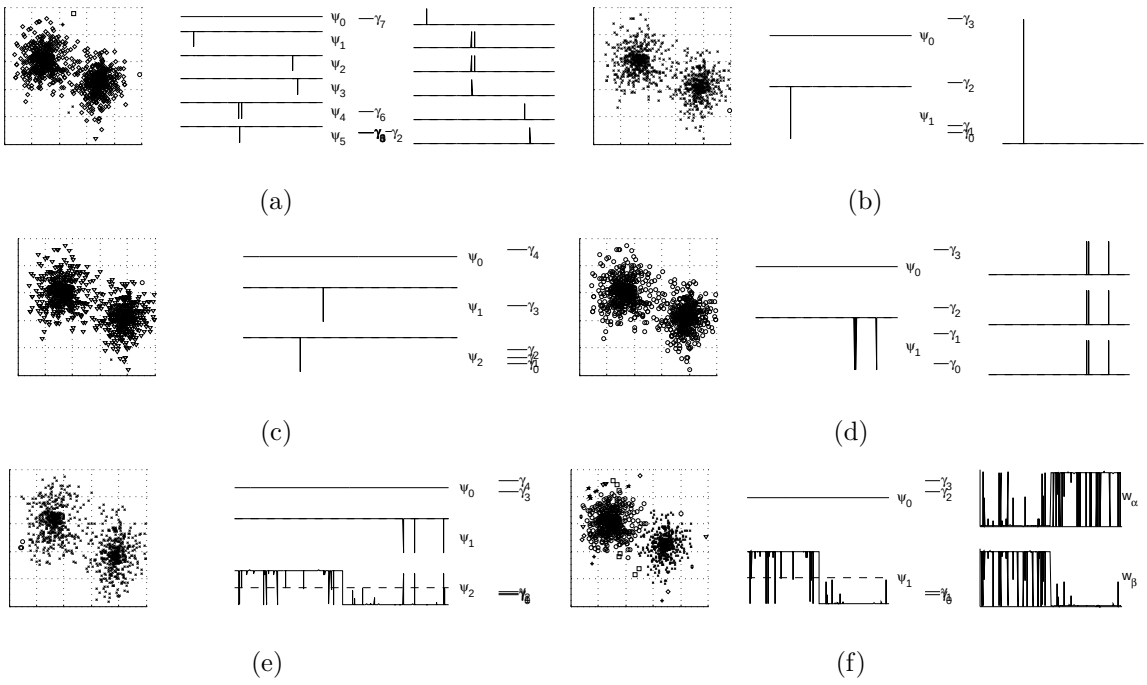


FIG. 10: Clouds data set clustered with outlier removal. Five iterations of outlier removal (a)-(e) detect outliers, remove them from the data set, and re-analyze the system. A structure reflecting the two clouds emerges in the final iteration (f), which is solved as an $m = 2$ problem. When outliers occur within the degenerate subspace, as in (a), (b), and (d), they are detected by constructing a linear combination of the degenerate eigenvectors. When two or more items are part of the same outlier, the linear combination constructed for each should be the identical. Hence, the two-member outlier in (a) and the three-member outlier in (d) give rise to redundant linear combinations. Outliers are detected within the non-degenerate subspace in (c) and (e).

The Intersecting (see Figure 12) and Target (see Figure 13) data sets similarly have outliers that may be removed directly or as isolated subsets. For both data sets, the algorithm was forced to remove outliers despite the presence of non-outlier low-lying eigenvectors. Intersecting may be solved as an $m = 4$ problem, as in Figure 12(a), or as an $m = 2$ problem after outliers are first removed from the non-degenerate space, as in Figure 12(b). The assignment functions $\boldsymbol{w}_\alpha$ and $\boldsymbol{w}_\beta$ were constructed from Eq. (26) in the former case, but directly created during outlier detection in the latter. The outliers at the corners of the Target data set are extracted from the degenerate subspace of Figure 13(a) to reveal the $m = 2$ clustering of Figure 13(b). The data set is analyzed as an $m = 6$ problem in Sec. IV.

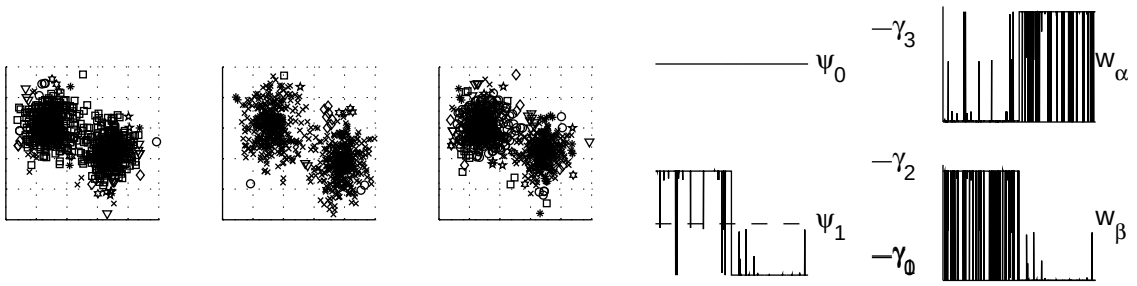


FIG. 11: Clouds data set clustered by removing outliers past the first eigenspectrum gap. Outlier removal continues as long as at least $m-2$ low-lying eigenvectors exhibit an outlier signature. Two iterations remove a total of 80 items within outliers, as shown in the first two data set plots. The final plot shows the $m = 2$ clustering that becomes possible after removing the obfuscating items. The resulting eigensystem and non-outlier assignment functions are displayed.

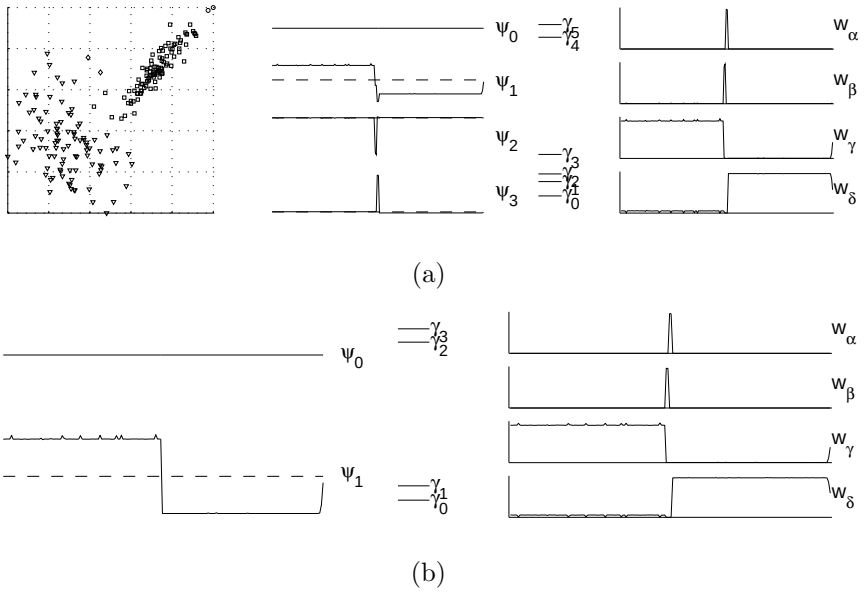


FIG. 12: Intersecting data set. Outliers are manifest in eigenvectors $\boldsymbol{\psi}_2$ and $\boldsymbol{\psi}_3$, within the non-degenerate subspace. (a) Clustered as an $m = 4$ problem. (b) Clustered as an $m = 2$ problem after first removing the outliers in a pre-processing step. The outliers are removed and the system is re-analyzed to yield the eigensystem shown in (b).

For the three data sets above, analysis with and without outlier detection yields clusters of similar certainty. Figures 14–16 compare the certainties of non-outlier clusters derived using several methods for handling outliers. If outliers are not given special consideration, they are nevertheless removed as isolated subsets—small, well-defined clusters. Since this

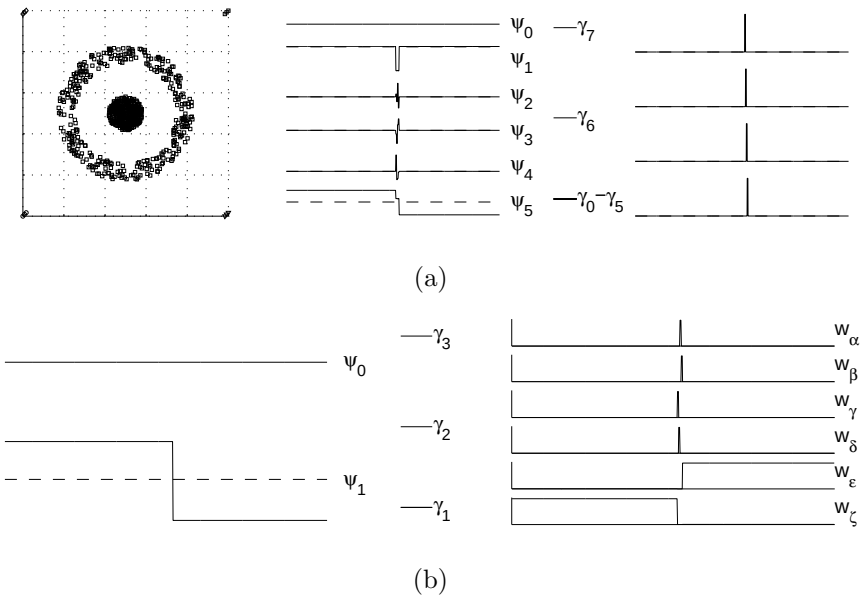


FIG. 13: Target data set. Outliers are manifest in eigenvectors $\boldsymbol{\psi}_1 - \boldsymbol{\psi}_4$, within the degenerate subspace. (a) Shown are the target data set, eigenvectors, eigenvalues, and linear combinations of eigenvectors within the degenerate subspace that signal outliers. (b) The outliers reflected in the linear combinations from (a) are removed, yielding the displayed eigensystem. Macrostate data clustering computes assignment functions $\boldsymbol{w}_\epsilon$ and $\boldsymbol{w}_\zeta$ from $\boldsymbol{\psi}_0$ and $\boldsymbol{\psi}_1$; each of the $\boldsymbol{w}_\alpha - \boldsymbol{w}_\delta$ were constructed for a particular outlier.

method does not differentiate outliers from other clusters, it calculates outlier certainties; both conditional and unconditional certainties are reported for the one recursively-analyzed data set, Clouds, where the distinction is relevant. The two other procedures directly detect outliers using outlier signatures. The first search is restricted from crossing the first gap in the eigenspectrum, while the second is not and instead searches the first $f_{\text{range}} * N$ eigenvectors for outliers. Certainties for the Clouds data set show mild improvement when outliers beyond the first eigenspectrum gap are detected, likely due to the increased number of outliers removed. For the smaller Intersecting and Target problems, the few outlying items that remain in the low-lying eigenvectors under the isolated subset approach do not result in appreciably different certainties than those derived from eigenvectors free of outliers.

	$\overline{\Upsilon}_\alpha(M)$	$\overline{\Upsilon}_\beta(M)$	Outlying items removed
Isolated subset (conditional)	0.968706	0.973189	23
Isolated subset (unconditional)	0.970969	0.975043	23
Outlier removal	0.973572	0.974218	36
Outlier removal beyond first gap	0.985971	0.986599	80

FIG. 14: Comparison of certainties for non-outlier clusters of Clouds data set. Outlying items removed refers to the total number of items belonging to an outlier that were removed either recursively or via detection.

	$\overline{\Upsilon}_\alpha(M)$	$\overline{\Upsilon}_\beta(M)$
Isolated subset	0.942132	0.949347
Outlier removal	0.944269	0.950097

FIG. 15: Comparison of certainties for non-outlier clusters of Intersecting data set.

E. Computational Implementation

Figures 17-20 describe the clustering process as a flow of execution between modules that have previously been discussed. The highest-level overview, presented by Figure 17, shows that clustering begins with a definition of Γ, whose eigenanalysis results in the (left) eigenvectors $\boldsymbol{\varphi}_n$ and eigenvalues γ_n that are subsequently used to determine the number of clusters m and the assignment functions $\boldsymbol{w}_\alpha$. If the data set is amenable to clustering, i.e., $m > 1$, then any item i that is strongly identified with a cluster α is assigned to a sub-problem characterized by the transition matrix Γ_α. This transition matrix is then recursively clustered, as discussed in greater detail in Sec. III C. Items that are not strongly assigned to

	$\overline{\Upsilon}_\alpha(M)$	$\overline{\Upsilon}_\beta(M)$
Isolated subset	0.999954	0.999959
Outlier removal	0.999994	0.999994

FIG. 16: Comparison of certainties for non-outlier clusters of Target data set.

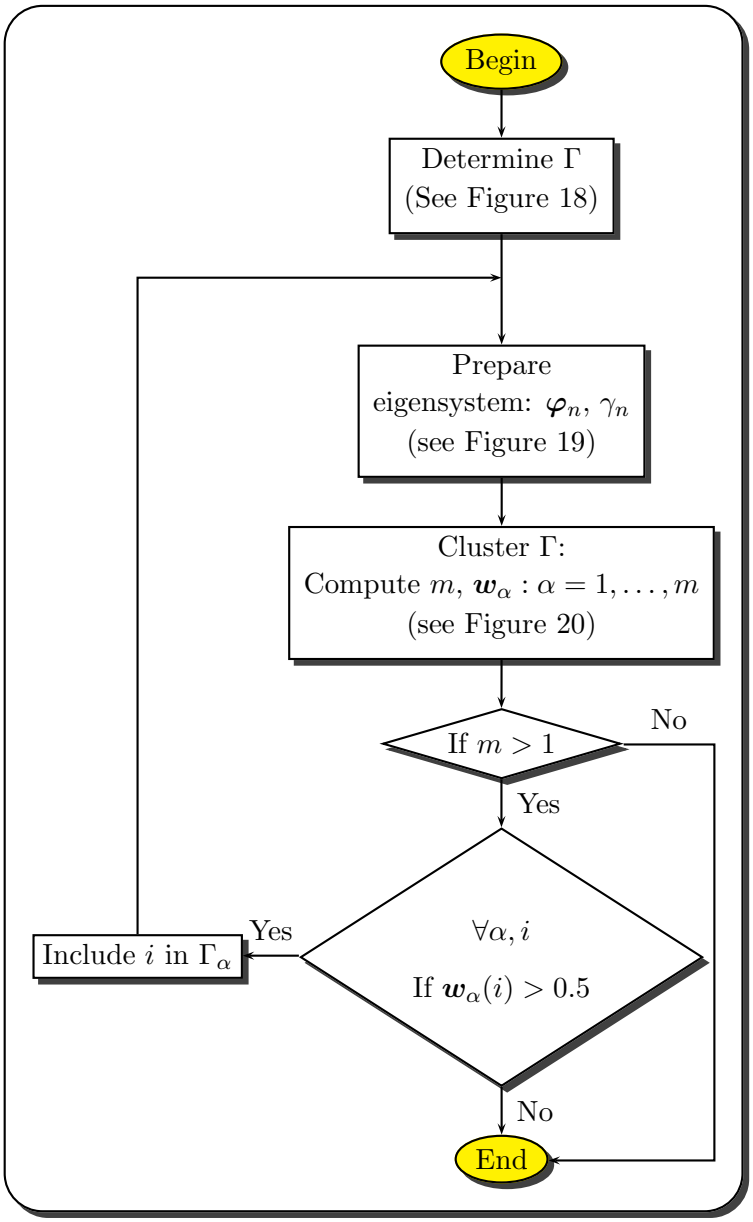


FIG. 17: Overview of recursive macrostate data clustering with hard, majority-based assignments.

a single cluster are not recursively examined: their fuzzy, relative assignment probabilities are described by those $\boldsymbol{w}_\alpha$ already constructed. Figures 18-20 provide additional detail concerning the first three steps in the clustering process.

Computation and conditioning of Γ is outlined in Figure 18. For data sets described in terms of measurements X, the dissimilarity matrix D is computed from the data after the dimensions are scaled by the problem-specific metric tensor g. In many domains, dissimilarities are not a simple function of the input and so are not readily computed via Eq.

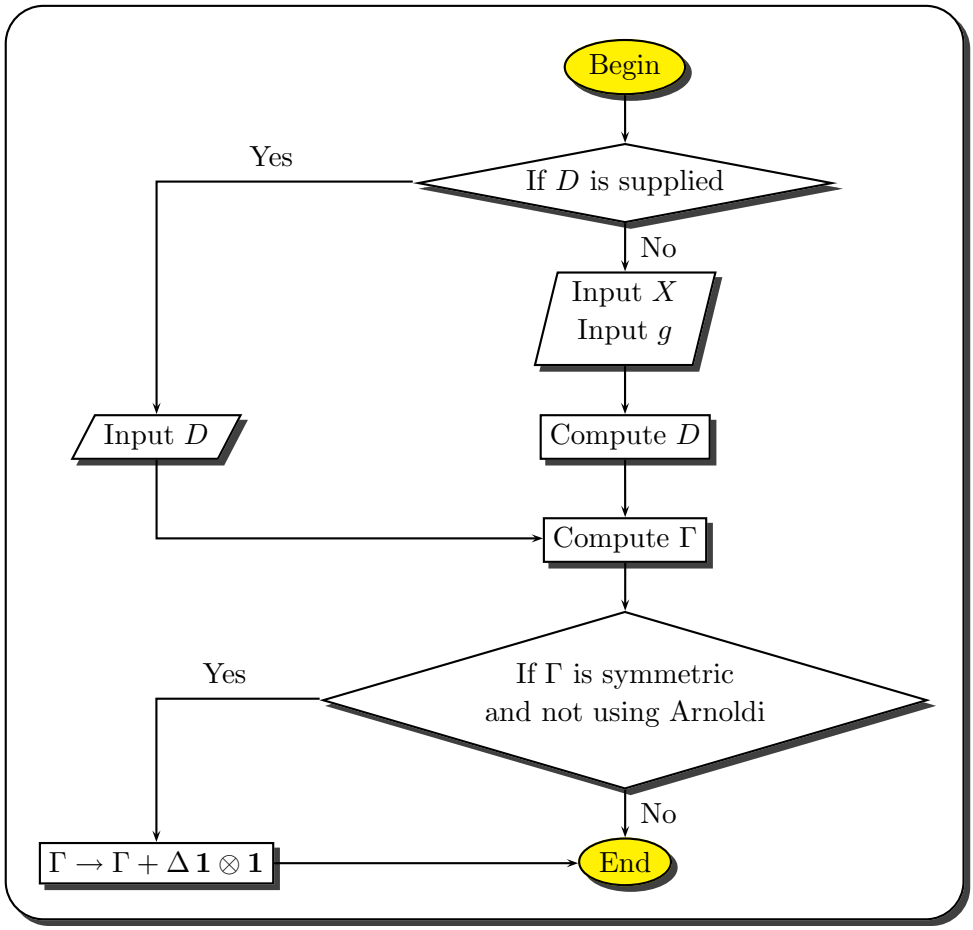


FIG. 18: Preparation of Γ.

(1). For example, the raw protein sequences input to a sequence alignment problem may be compared using dynamic programming techniques to produce pairwise similarities. This processing is best performed externally to the clustering algorithm, which directly uses the D matrix computed offline. Regardless of how D is defined, Γ is computed from it via Eq. (18) and then shifted by the outer product of $\mathbf{1}$ with itself to guarantee that $\boldsymbol{\varphi}_0 = \mathbf{1}$. This last step is only valid when Γ is symmetric. The algorithm may be configured to use the Arnoldi method [56] to compute a subset of the eigenspectrum. In this case it is best not to shift Γ, as this will artificially introduce a gap between γ_0 and the other low-lying eigenvalues that would complicate convergence. When the Arnoldi method is used or when Γ is not symmetric, $\boldsymbol{\varphi}_0 = \mathbf{1}$ may be forced by a subsequent phase using the method described in Sec. II D.

The potentially iterative process of computing the eigensystem of the Γ computed by Fig-

ure 18 and removing its outliers is shown in Figure 19. After the eigensystem is computed the procedures discussed in Sec. III D are used to remove outliers detected in the degenerate and non-degenerate subspaces. If any outliers are removed, the eigensystem is recomputed. Outlier detection in the degenerate subspace examines a linear combination $\boldsymbol{\psi}_{\text{outlier}}^{\text{degenerate}}$ of the degenerate eigenvectors created for each of the $O(N)$ items i. In contrast, within the non-degenerate subspace the vectors examined are the non-degenerate eigenvectors whose corresponding eigenvalues precede the first eigenspectrum gap or whose index is less than $f_{\text{range}} * N$. Once the candidate vector is determined, it is processed similarly within each subspace. In particular, its $O(N)$ squared components are collected and sorted, requiring $O(N \log N)$ operations, so that the largest $N_{\text{outlier}}^{\max}$ squared components may be summed and compared to the threshold $f_{\text{outlier}} * N$. Therefore, outlier detection within the degenerate subspace requires $O(N^2 \log N)$ operations, while within the non-degenerate subspace it requires $O(f_{\text{range}} * N^2 \log N) = O(N \log N)$, assuming that $f_{\text{range}} * N$ is a small constant.

Computing the eigensystem of Γ is the most computationally expensive step in the clustering process. Approaches to computing the entire eigenspectrum of a dense matrix require $O(N^3)$ operations, though the constant prefactors may vary considerably between algorithms. For dense, symmetric systems, the current implementation uses LAPACK's [57] `dsyev`, which reduces Γ to tridiagonal form before using the implicit QR algorithm to determine its eigensystem (see Sec. 8.3 of Ref. [54]). `dsyevr` [58] is a more efficient alternative, but requires additional memory. The implementation solves asymmetric systems using `dgeev` [59], which reduces Γ to upper Hessenberg form, computes a Schur decomposition using the implicit QR algorithm, and finally computes the eigenvectors of an upper quasi-triangular matrix (see Sec. 7.5.6 of Ref. [54]).

Greater efficiency may be achieved by using the Arnoldi method, which reduces to the Lanczos method for symmetric matrices, to compute only the small, relevant subset of the eigenspectrum near the zero eigenvalues. The Arnoldi method is an iterative process for computing the tridiagonalization of Γ and is attractive because extremal eigenvalues and their associated eigenvectors often emerge long before tridiagonalization is complete (see Ref. [60] and Chapter 9 of Ref. [54]). Computation within an iterative step is dominated by a matrix-vector multiplication involving Γ, which requires $O(N^2)$ operations if Γ is dense, but only $O(i * N)$ if Γ has a sparse representation and on average i non-zeros per row.

Unfortunately, the Arnoldi method has poor convergence properties for the small, densely-

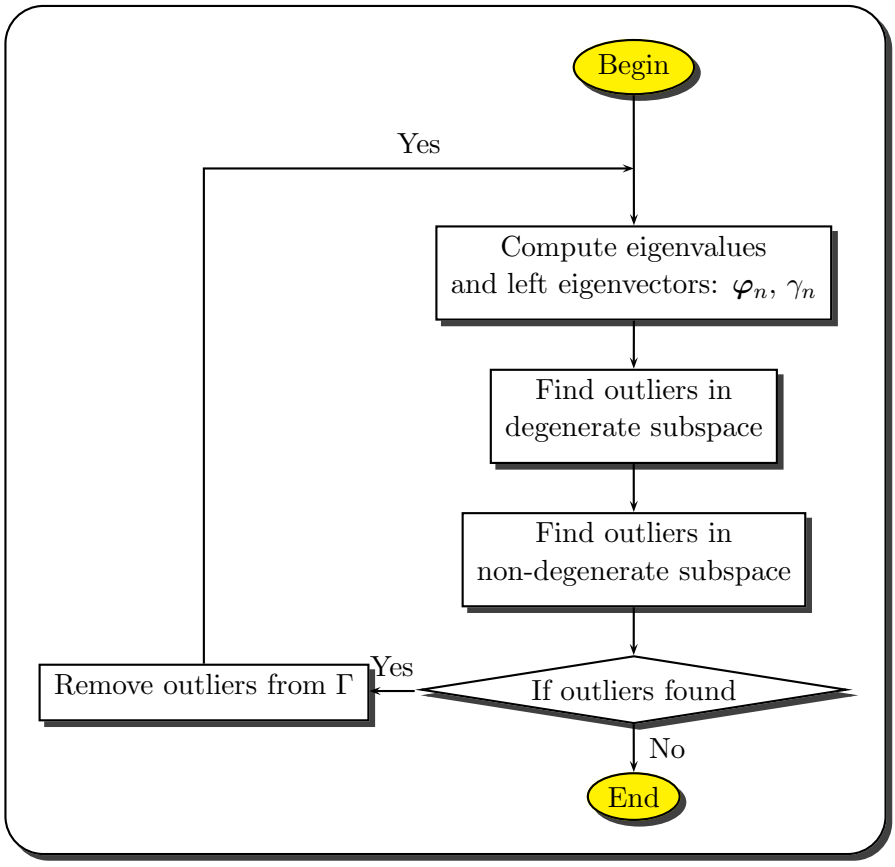


FIG. 19: Outlier removal and preparation of eigensystem.

packed eigenvalues required for macrostate data clustering. The related eigenvalue problem

$$(\Gamma - \sigma\, I)^{-1}\, \boldsymbol{\varphi}_n = \nu_n\, \boldsymbol{\varphi}_n \ ,$$

where $\nu_n = (\gamma_n - \sigma)^{-1}$, defines a shift-and-invert spectral transformation [61]. This procedure aids convergence of those eigenvalues near σ, which will have an eigenvalue ν_n of large magnitude in the transformed problem. The current implementation uses a near-zero σ of $\sqrt{\epsilon}$, where ϵ is machine precision. It is based on the ARPACK++ [62] package that provides C++ wrappers around ARPACK [63] routines.

The core of the recursive clustering algorithm is the construction of the assignment functions via Eq. (26), where the orthogonal vectors $\hat{\boldsymbol{\varphi}}_n$ generalize the use of eigenvectors $\boldsymbol{\psi}_n$ in the symmetric case. The process depicted in Figure 20 replaces the original brute-force method for performing macrostate data clustering used by Korenblum and Shalloway [5]. When Γ is asymmetric or the Arnoldi method is used to compute only the low-lying end of the eigenspectrum, $\boldsymbol{\varphi}_0$ will not have been forced to $\mathbf{1}$ in Figure 18. In fact, if Γ is asymmet-

ric, the left eigenvectors will not be mutually orthogonal, but are instead bi-orthogonal with respect to the right eigenvectors. In these cases, the procedure described in Sec. II D is used to compute an orthogonal subspace spanned by a set of vectors $\hat{\boldsymbol{\varphi}}_n$, with $\hat{\boldsymbol{\varphi}}_0 = \mathbf{1}$. Otherwise, the (left) eigenvectors are used directly as the $\hat{\boldsymbol{\varphi}}_n$. Forming the orthogonal subspace requires the singular value decomposition of an $N \times m$ matrix A of the m low-lying eigenvectors, which may be computed in $O(Nm^2 + m^3)$ operations (see Sec. 5.4.5 of Ref. [54]). The current implementation computes the SVD using `dgesvd`.

An approximate solution μ is computed from the m orthogonal vectors $\hat{\boldsymbol{\varphi}}_n$ by a process schematically represented in Figure 21. A linear program imposes the constraints of Eqs. (27) and (28) on μ to determine the exact solution M, whose elements serve as the expansion coefficients used to define the $\boldsymbol{w}_\alpha$. The constrained linear program is solved using the GLPK [64] implementation of the simplex method. As discussed in Sec. III B, the simplex method is theoretically exponential, but efficient in practice. In particular, it has proven to be considerably more efficient than the eigensystem solver, such that the latter remains the computational bottleneck.

Determining the approximate solution μ requires finding m representatives and then setting their assignment intensities to zero or unity by solving m linear systems, as discussed in Sec. III A. Representative selection uses $O(N^2 + m^3 N)$ operations: an $O(N^2)$ comparison is used to determine the two mutually furthest representatives, which seed the set; the remaining $m-2$ representatives are chosen from the $O(N)$ items that have not already been selected as representatives by comparing and computing the distance and sign structure of each, using $O(m)$ operations, with respect to the $O(m)$ existing representatives. Each of the m linear systems is computed using the $O(m^2)$ `dgesv` algorithm. It computes the solution to a real system of linear equations $A * X = B$ by using an LU decomposition with partial pivoting and row interchange to factor A as $A = P\,L\,U$, where P is a permutation matrix, L is unit lower triangular, and U is upper triangular. If $A \in \mathbb{R}^{m \times m}$, then factorizing A requires $O(m^2)$ operations (see Sec. 3.4.3 of Ref. [54]). Forward and backward substitution then reduce the factored form to a solution using $O(m^2)$ additional operations each (see Sec. 3.1 of Ref. [54]).

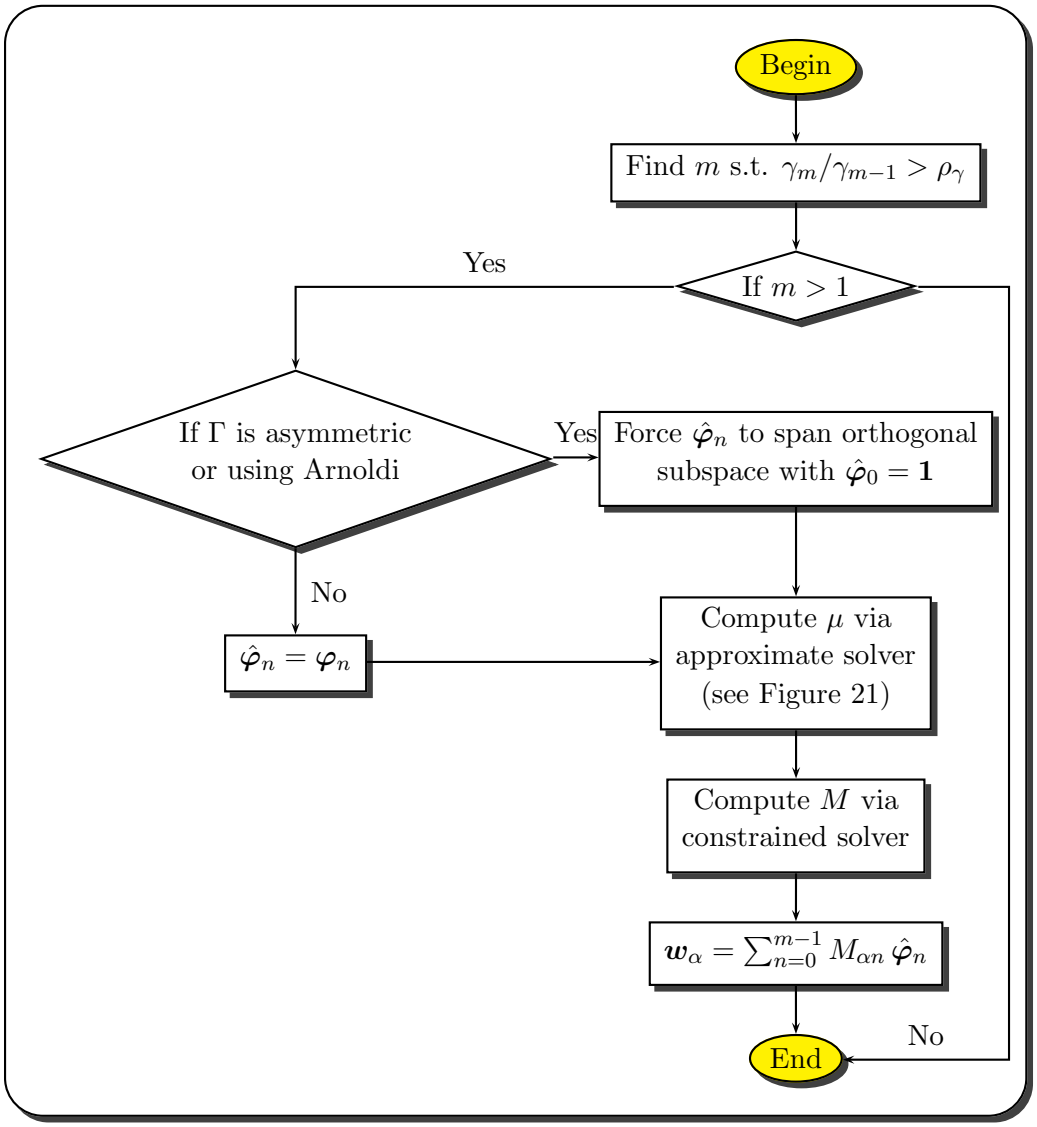


FIG. 20: Fuzzy macrostate data clustering.

IV. RESULTS

In their evaluation of macrostate data clustering, Korenblum and Shalloway presented a series of problems that had challenged traditional clustering methods, such as k-means and agglomerative clustering. To show that macrostate data clustering does not rely on low dimensionality, they further considered, and successfully clustered, items embedded in a 20 dimensional space. The present evaluation extends these results by considering data sets from the Fundamental Clustering Problems Suite (FCPS) [2] in Sec. IV A. Macrostate data clustering reproduces the author's results for all but the Engy Time data set, though the

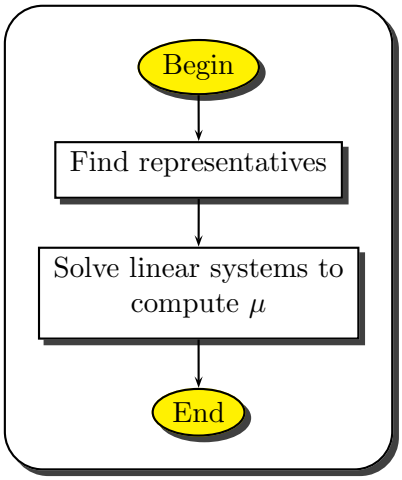


FIG. 21: Approximate solver.

proposed differentiation into two clusters appears debatable. The scalability of macrostate data clustering is demonstrated in Sec. IV B, which varies N and m to consider problems as large as 20,000 items grouped in nine clusters.

The problems analyzed in this section, and throughout Sec. III, define the dissimilarity matrix D directly from the raw coordinate data using Eq. (1) with $g_{ab} = \delta_{ab}$. The minimum gap parameter ρ_γ was set to 3.

The current implementation is written in C++, though it makes use of libraries written in C and Fortran. Access to low-level LAPACK [57] routines was provided by version 2.3.0 of the LAPACK++ [65] C++ wrappers. The Arnoldi eigensystem solver was implemented via ARPACK [63] routines and exposed through the ARPACK++ [62] C++ wrappers. The implementation of the simplex method for solving constrained linear programs was provided by GLPK [64] version 4.9. The clustering application and its dependences were compiled using `gcc/g++` version 4.1.2 and `gfortran` version 4.1.2; both were passed the `-O3` optimization flag. The scaling results of Sec. IV B were executed on a dedicated quad CPU 3.46 GHz Pentium 4 node, configured with 4GB of RAM and 4GB of swap space, and running a 64-bit version of SuSE Linux.

A. Bi- and Tri-variate Test Cases

The Fundamental Clustering Problem Suite (FCPS) [2] was developed as a benchmark for clustering algorithms and is distributed with known classifications. The suite contains problems that can not be clustered by k-means and agglomerative methods. Since Sec. I presents the Crescentric data set [4] as one problematic to complete linkage and k-means, its proper classification by macrostate data clustering is presented here as well. The analysis of Crescentric and nine of the ten FCPS data sets are displayed in Figures 22 and 23, which show that macrostate data clustering reproduces the subjective clustering for each of them. The final FCPS data set, Engy Time, is considered separately in Figure 24 since the results differ from those suggested by Ultsch [2].

The data sets consist of two or three measurements N_M on each of the N items. Therefore, the items may be represented as N points in an N_M-dimensional space. For example, the items of the Atom data set of Figure 22(e) have $N_M = 3$ measurements and so are embedded in a three-dimensional space. Half of the $N = 800$ are tightly grouped within the core of a globe comprised of the remaining items, so that the variance within the core is significantly smaller than that on the surface of the globe. The data set can not be clustered by k-means since it is not linearly separable. The figure shows the two low-lying eigenvectors corresponding to the eigenvalues preceding the first eigenspectrum gap that separates eigenvalues γ_1 and γ_2. Through this gap the algorithm infers that the data set is properly dissected into $m = 2$ clusters; there is no gap, for example, between eigenvalues γ_2 and γ_3 and an $m = 3$ clustering would overdissect the space. As expected, $\boldsymbol{\psi}_0$ is constant since Eq. (25) is enforced through the procedure of Appendix B, when Γ is symmetric, or Sec. II D, when Γ is asymmetric. Since the items were arranged according to their subjective cluster, the presence of the two equally-sized clusters is clearly recognizable in the step-function form of $\boldsymbol{\psi}_1$.

The degeneracy of the zero eigenvalue indicates isolated subsets, which leads to the perfectly level structure of $\boldsymbol{\psi}_1$. This allows construction of level assignment functions, through which items are assigned to clusters with probabilities strictly equal to zero or unity. Therefore, there is no uncertainty and the fuzzy clustering approach has determined a hard clustering. For each problem, Figure 23 displays the number of clusters m, the magnitude of the spectral gap γ_m/γ_{m-1}, and the certainty $\overline{\Upsilon}_\alpha(M)$ and range of assignment intensities for

each cluster within the problem. The assignment intensities are the values of the $\boldsymbol{w}_\alpha(i)$ for those items i subjectively assigned to cluster α. An interesting cluster requires $m > 1$ so that the denominator of the spectral gap ratio is generally non-zero and the ratio is finite. However, for degenerate problems, such as Atom, the denominator is always zero and the ratio is reported as ∞. Such degenerate problems lead to hard clusterings, in which each cluster enjoys absolute certainty and all of the assignment intensities are unity. To avoid redundancy in the figure, such m-way degenerate problems have only a single entry indicating that the certainties of *all* clusters are unity; the assignment intensities are not individually listed for each cluster either, but are indicated as arising from a hard clustering.

The Tetra data set, depicted in Figure 22(c), has four clusters arranged at the corners of a tetrahedron such that they nearly overlap. Despite their relative proximity, the $m = 4$ clusters are recognizable from the gap in the eigenspectrum. However, unlike Atom, the problem is not degenerate. The result is a perturbation in the level structure of the eigenvectors, as discussed in Sec. II C, which is translated to fuzziness in the assignment functions. Figure 23 indicates that perturbation leads to assignment intensities as low as 0.55 and markedly lower certainties, though the corresponding clusters remain subjectively correct.

Of the remaining problems, Hepta, Lsun, Chainlink, and Target induce hard clusterings, though the solution to Wing Nut is near certain. Hepta has seven well-defined clusters, six of which surround, in three dimensions, a smaller seventh cluster. The three two-dimensional clusters of Lsun, two of which are rectangular and nearly perpendicular, were designed to test an algorithm's ability to cope with different intra-cluster variances and inter-cluster separations. Like Atom, a proper solution to Chainlink requires differentiating clusters that are not linearly separable: its two rings interlock in three dimensions. Target consists of six clusters, two of which are concentric circles and the remainder of which are outliers scattered to the four extreme corners of its two-dimensional space. Target is solved in Figure 22(f) as an $m = 6$ problem, in which the outliers are handled analogously to the two larger clusters; the four outliers were detected and removed directly in Sec. III D.

Only Two Diamonds and Crescentric exhibit a level of uncertainty commensurate to the fuzziness of Tetra. The two clusters of the Two Diamonds data set abut such that items near the interface separating the two diamonds are assigned relatively low intensities. Similar ambiguity exists in the Crescentric data set, in which two crescents are closely

juxtaposed: an item projecting out towards an opposing cluster has some affinity for that cluster, which detracts from its probability of assignment to its own subjective cluster. This reflects the expectation that assignment functions will be most fuzzy near the boundary between clusters. The described weakly-assigned items have a strong correlation with those misclassified by distance-based clustering schemes in Figure 1.

Golf Ball is a compact sphere of items that can not be subjectively clustered. Hence, its eigenvectors are unstructured and the only spectral gap separates γ_0 from γ_1, indicating that there is only a single monolithic cluster. Korenblum and Shalloway found a similar result when analyzing a random data set: macrostate data clustering is not mislead into suggesting a spurious clustering when none exists.

The final FCPS data set, Engy Time, is a two-dimensional mixture of two Gaussian distributions. A clustering approach based on self-organizing maps [2] used distance and density relationships to differentiate the two Gaussian structures. While the different distribution widths make two clusters subjectively visible, their strong connectivity mean that items along the boundary, but subjectively assigned to different clusters, will be highly correlated. Therefore, the determination by macrostate data clustering that no clustering is possible appears to be valid, if not preferable.

B. Scaling Benchmarks

Macrostate data clustering scales to problem sizes of biological interest: it solves a problem with $N = 20,000$ items in under an hour and a half. The scaling results obtained from applying macrostate data clustering to synthetic data sets are shown in Figure 25. Each of the curves corresponds to a different number of clusters m, while the x-axis describes the total number of items N across all clusters. The y-axis gives the time in seconds to cluster the corresponding data set, which is composed of m two-dimensional clusters surrounding a common center of mass. Optimization times are dominated by the time spent in the numerical eigensystem solver. Our use of an iterative Arnoldi solver, whose execution time is dependent on both the size and configuration of the data set, likely explains the spikes in the

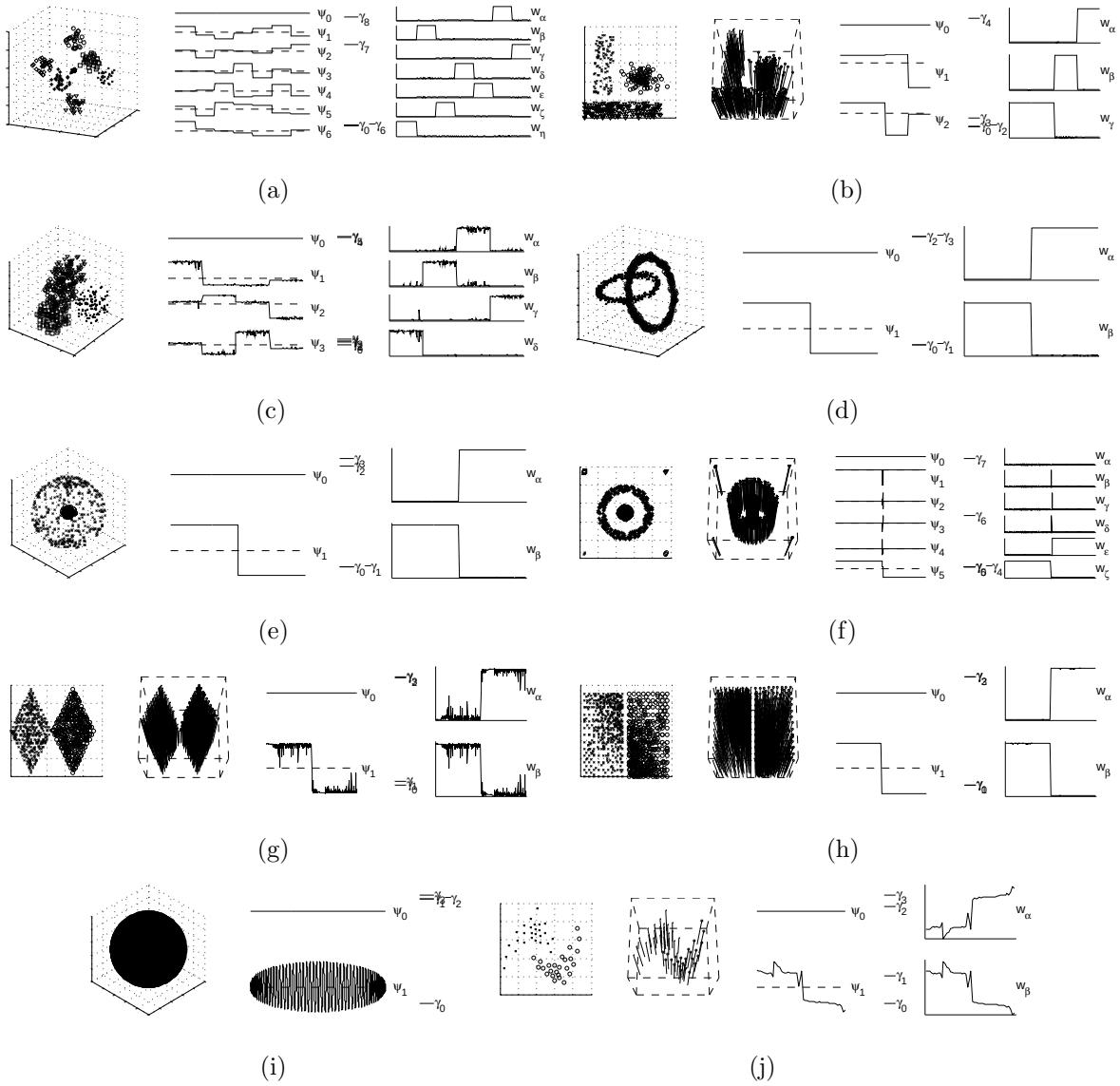


FIG. 22: Bi- and tri-variate test cases. (a) Hepta (b) Lsun (c) Tetra (d) Chainlink (e) Atom (f) Target (g) Two Diamonds (h) Wing Nut (i) Golf Ball (j) Crescentric.

curves. Each problem was solved five times, with little variance between runs. The figure shows a clear power-law relation between N and execution time. The average exponent of the three smoothest curves—those for $m = 7, 8$, and 9—is 3.1, consistent with the expected N^3 scaling of a numerical eigensolver.

Problem	m	$\frac{\gamma_m}{\gamma_{m-1}}$	$\overline{\Upsilon}_\alpha(M)$	Assignment	Problem	m	$\frac{\gamma_m}{\gamma_{m-1}}$	$\overline{\Upsilon}_\alpha(M)$	Assignment
Hepta	7	∞	1.00	Hard Clustering	Target	6		1.00	Hard Clustering
Lsun	3	∞	1.00	Hard Clustering	Two Diamonds	2	29.31	0.93	0.59-1.00
Tetra	4	17.21	0.87	0.74-1.00				0.93	0.53-1.00
			0.90	0.77-1.00	Wing Nut	2	245.95	1.00	0.99-1.00
			0.91	0.87-1.00				0.99	0.99-1.00
			0.93	0.55-1.00	Golf Ball	1			
Chainlink	2	∞	1.00	Hard Clustering	Crescentric	2		0.71	0.75-1.00
Atom	2	∞	1.00	Hard Clustering				0.71	0.51-1.00

FIG. 23: Analysis of bi- and tri-variate test cases.

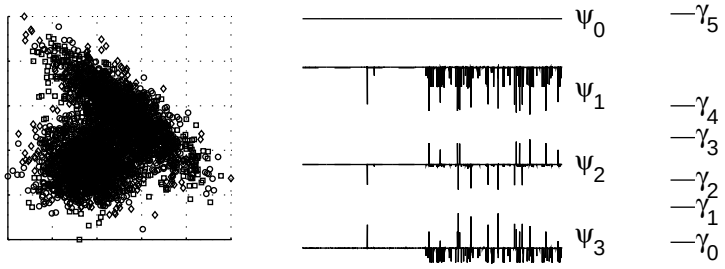


FIG. 24: Engy Time.

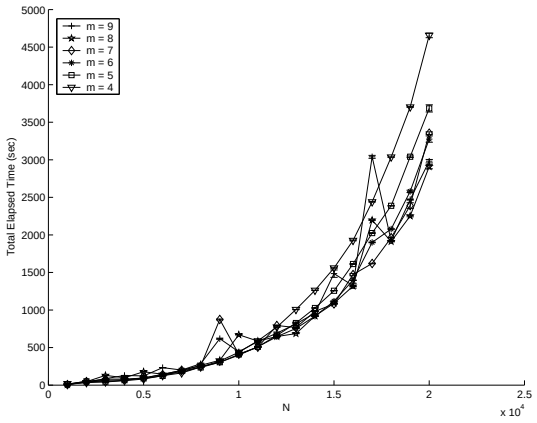


FIG. 25: Scaling results for synthetic benchmarks with N varied from $1,000$ to $20,000$ in steps of $1,000$ and with m varied from 4 to 9.

V. DISCUSSION

Macrostate data clustering has been developed as a fuzzy, partitional, and recursive method that performs well on problems that have challenged traditional, distance-based approaches, such as k-means and agglomerative clustering. Like other spectral methods, it succeeds where traditional methods have failed by exploiting global inter-item connectivity information preserved in the structure of eigenvectors of a system derived from a dissimilarity matrix D. The appropriate number of clusters is determined directly from the eigenspectrum gaps γ_m/γ_{m-1} and need not be specified *a priori*. The corresponding acceptance parameter ρ_γ was determined empirically by Korenblum and Shalloway [5] and has given good results for a broad range of test cases. The number of clusters m determines the number of low-lying eigenvectors to be used as a basis for the linear expansion $\boldsymbol{w}_\alpha$ that probabilistically describes membership of cluster α.

Practitioners have long understood that the number of clusters could be ascertained from a gap in the eigenspectrum [7, 10, 22, 23, 24, 25, 33, 34]. However, prior to the gap condition suggested by Korenblum and Shalloway [5], they have resorted to manual inspection. Zelnik-Manor and Perona [26] instead determine m as the number of low-lying eigenvectors of a normalized affinity matrix, which, when rotated, best approximate a block-diagonal matrix.

Eigenvectors have previously been used in the clustering process. The most straightforward approaches, such as recursive spectral bipartitioning [13], threshold the single Fiedler vector at each stage and do not require knowledge of the number of clusters. However, several experimental [10, 66] and theoretical [29] studies have shown that direct m-way partitioning may yield better results than recursive bipartitioning. Occasionally these non-hierarchical, partitional schemes decouple the number of clusters m from the number of eigenvectors used to determine them, e.g., using as many eigenvectors as practically possible [9]. However, in most cases, m determines both the number of clusters and the number of low-lying eigenvectors used to express them.

m-way partitioning schemes frequently make use of an eigenspace representation, wherein item i takes the form of an m-vector $\overrightarrow{\boldsymbol{\psi}}_i$ comprised of the i^{th} components of the first m low-lying eigenvectors. A hard clustering of these item vectors may be obtained using k-means [7, 24], based on the sign of each component [23, 25], or according to the vector's direction [10, 22] and magnitude [9]. The $\boldsymbol{w}_\alpha$ may also take on real values. In a bipartitioning

context, the Fiedler vector assigns a continuous weight within a bounded interval to an item, which associates it with one of the two clusters [67]. Drineas et al. [68] have used the eigenvectors as the $\boldsymbol{w}_\alpha$ directly, though the interpretation as intensities is loosely defined.

The primary feature distinguishing macrostate data clustering, as developed by Korenblum and Shalloway [5], from other real-valued, m-way partitioning approaches is the interpretation of the $\boldsymbol{w}_\alpha(i)$ as the probability that item i is assigned to cluster α. This probabilistic interpretation requires that the assignment functions be expressed as a linear combination of low-lying eigenvectors, which generalizes the eigenspace representation $\overrightarrow{\boldsymbol{\psi}}_i$ of item i. The inherent fuzziness of the resulting $\boldsymbol{w}_\alpha$ introduces cluster overlap and an attendant uncertainty, which leads naturally to an information-theoretic objective function and the principle of uncertainty minimization. It is this principle, absent in related work lacking a probabilistic interpretation, that ultimately determines the expansion coefficients of the linear combination.

Components of the macrostate data clustering solver may be valuable even for those domains in which a hard clustering suffices, such as molecular dynamics. The validity of a probabilistic interpretation is ensured by the constrained solver. However, as discussed in Sec. III B, the solution M determined by the constrained solver is expected to be a small perturbation of the solution μ found by the approximate solver. Therefore, thresholding the assignment functions $\boldsymbol{w}_\alpha$ computed from μ is likely to produce the same subjective clustering as that computed from M, without the additional algorithmic and run-time overhead of the constrained solver.

Hard clustering has recently been used to improve the computational efficiency of molecular dynamics studies of protein folding [69]. Characterizing folding rates and trajectories through configuration space is informative in elucidating pathways and in studying diseases caused by misfolded proteins. Unfortunately, experimental observations frequently can not isolate individual protein states, but instead present ensemble averages. While *in silico* studies offer atomistic detail of a single trajectory, the system's fast vibrational modes restrict integration time steps to femtoseconds and hence severely limit the study of significant folding events occurring at microsecond granularity [70]. Since it is the transitions between the metastable conformations that are of physical interest, rather than the high-frequency, intra-macrostate fluctuations, Chodera et al. [69] have proposed coarse graining configuration space to a set of macrostates and then initiating short simulation trajectories from

each in order to establish a Markov model. As the model describes the transitions between the physically-related states, the authors suggest that it provides a practical alternative to long, fine-grained simulations for deriving properties such as state lifetimes [71] and mean first-passage times [72].

Shalloway [43] has previously described an analytic approach for dissecting configuration space based on the Gibbs-Boltzmann distribution that could provide the required coarse graining. This was tersely summarized in Sec. II A. Schütte et al. [73] have proposed an alternative approach that effectively constructs the transition matrix Γ from Monte Carlo simulation. A variant of parallel tempering [74] allows efficient sampling of configuration space in the presence of energy barriers. Eigenanalysis of Γ and clustering of the resulting $\overrightarrow{\psi}_i$ according to sign structure [23] lead to a characterization of macrostates suitable for the kinetic model of Chodera et al.

Macrostate data clustering has benefited from the sign structures described by Deuflhard et al. [23] and there are strong parallels between this work and theirs. However, as discussed in Sec. III A, the sign structures used to segregate microstates into macrostates may have ambiguities caused by numerical imprecision. Deuflhard et al. resolved this issue by assigning ambiguous microstates to macrostates using a least squares procedure. As discussed in Appendix C, the use of sign structures within macrostate data clustering is restricted to differentiating (potential) representatives, whose sign structure should be well defined. Therefore, the approximate solver may provide a more robust and straightforward alternative to determining the macrostates of a Γ matrix defined from molecular dynamics experiments. Since the macrostates are necessarily a non-fuzzy partition of the state space, the constrained solver need not be invoked; rather, the fuzzy assignment functions derived from the approximate solver should be thresholded to determine the macrostates.

When an individual is the ultimate consumer of the clustering results, a fuzzy approach is more informative than a hard clustering. In this case, relatively low assignment probabilities indicate that the corresponding items deserve special attention, while those assigned with high probability may be quickly verified or trusted outright. Manually-curated databases, such as the structural classification of proteins (SCOP) database [39], are a compelling application for fuzzy clustering. Using structural and evolutionary information, domain experts locate a protein domain within the SCOP hierarchy describing, from least to most constraining, its class, fold, superfamily, and family. Domains within the same superfamily

are believed to be related evolutionarily.

The consideration afforded each domain leads to a significant lag time between the addition of a structure to the the PDB [75] and its classification within SCOP. The manual curation process [76] is aided by automatic methods. Nevertheless, as of February 20, 2007, the PDB contained 41,814 entries, while the most recent distribution of SCOP contained only 27,599 PDB entries. This release fully captured the PDB as of January 18, 2005– a lag of nearly two years at the time of its distribution in October, 2006. In general, SCOP distributions occur at most every several months and, often, much less frequently. Therefore, researchers may be forced to wait a considerable amount of time until a structure classification is made available.

Fuzzy clustering may be able to alleviate the lengthy curation process. Paccanaro et al. [3] have already shown that a spectral clustering method can faithfully reproduce many of the superfamily classifications from a subset of SCOP. Macrostate data clustering provides the additional benefit of indicating the confidence of a particular classification. Thus, misclassifications should be reflected by relatively low assignment probabilities, which would signal, either to a curator or to a researcher attempting to extend SCOP with a new structure, that a structure deserves manual consideration. Additionally, the ability of macrostate data clustering to recursively analyze clusters should aid it in discovering the families constituting a superfamily.

Applying macrostate data clustering to SCOP requires a suitable notion of dissimilarity. Paccanaro et al. defined distances in terms of the E-values returned by the BLAST [49] sequence comparison algorithm. Other sensible options include the more sensitive values returned by PSI-BLAST [51] or the tm-scores [77] resulting from direct structural alignments. When distances are derived from the measurement matrix X, this freedom is reflected in the definition of the metric tensor g, which may be used to scale data dimensions. Whether computed directly or derived from X, determining D requires domain expertise.

APPENDIX A: NUMERICAL PRECISION OF EIGENVALUES

Macrostate data clustering requires that an eigenspectrum be reliably partitioned into a degenerate space of zero eigenvalues, a low-lying, non-degenerate range of eigenvalues lying beneath the first spectral gap, and the remaining high-frequency end of the spectrum.

Where the distinction between degenerate and low-lying, non-degenerate eigenvalues is not necessary, all are referred to as low-lying eigenvalues. Differentiation between degenerate and non-degenerate spaces depends on accurately determining a zero eigenvalue despite numerical imprecision. When there are multiple degeneracies, the boundary between the two spaces determines the number of clusters m. Therefore, applying the gap condition within the degenerate subspace would result in an improper determination of m. Similarly, a conservative approach to outlier detection must locate the gap separating degenerate and non-degenerate eigenvalues, beyond which outlier removal should not proceed.

Numerical routines frequently provide accuracy bounds that would allow, in principle, for proper determination of zero eigenvalues [57]. However, an implementation-independent approach is preferable. A robust approach is to compare the magnitudes of approximations to the same eigenvalue, computed using two different methods. If the eigenvalue is zero analytically, the magnitude of its approximations will be dominated by noise. Hence, each will effectively be a small random number and their normalized difference will be large.

The eigenvalue γ_n satisfying $\Gamma\,\boldsymbol{\psi}_n = \gamma_n\,\boldsymbol{\psi}_n$ is approximated by the γ'_n returned by the eigensystem solver. For a symmetric Γ, $\Gamma^2\,\boldsymbol{\psi}_n = \gamma_n^2\,\boldsymbol{\psi}_n$, for which a numerical solver would return γ''_n. Given infinite numerical precision, the analytic identity $\gamma'_n = \sqrt{\gamma''_n}$ would hold. If $\gamma_n = 0$, both γ'_n and γ''_n will be dominated by noise, but their intended approximation of zero will be signaled by their large relative difference

$$\left| \frac{\left(\gamma'_n - \sqrt{|\gamma''_n|}\right)}{\min(\gamma'_n, \sqrt{|\gamma''_n|})} \right| \gg 0 \; .$$

The above approach doubles the number of invocations to the numerical eigensystem solver and may be impractical for large systems. A variant of this approach takes the second approximation γ''_n to be the numerical value of the analytic identity

$$\langle \boldsymbol{\psi}_n | \Gamma | \boldsymbol{\psi}_n \rangle = \gamma_n \; ,$$

for symmetric Γ, or

$$\langle \boldsymbol{\varphi}_n | \Gamma | \boldsymbol{\psi}_n \rangle = \gamma_n \; ,$$

for asymmetric Γ.

APPENDIX B: DEGENERATE "ZERO" EIGENVALUES

Korenblum and Shalloway have shown that $\mathbf{1}$ is the sole stationary eigenvector of Γ in a (numerically) non-degenerate system (see Appendix B of Ref. [5]). If the system is truly degenerate, Γ will be reducible such that it may be brought to a block diagonal form through permutation. Each block along the diagonal represents an isolated subset or "invariant aggregate" [23] $\mathcal{S}$, with $\Gamma_{ij} = 0$ if $i \in \mathcal{S}$ and $j \notin \mathcal{S}$. If the system is comprised of nearly isolated subsets, numerical inaccuracies may prevent distinction between zero and the small eigenvalues that represent transitions between the subsets. The system will again appear to be degenerate and the $\boldsymbol{\psi}_0$ returned by a numerical eigensystem solver will not satisfy Eq. (25), though a linear combination of the approximately degenerate eigenvectors will sum to $\mathbf{1}$.

One resolution to this issue is the generalization of the equality constraints, as expressed in terms of $\hat{\varepsilon}_0$ rather than $\boldsymbol{e}_1$. A more elegant solution is to *enforce* Eq. (25) by breaking the degeneracy such that the eigenvector of the shifted eigenvalue is set to be $\mathbf{1}$. Since Γ is symmetric, updating it by the outer product of $\mathbf{1}$ with itself, according to

$$\Gamma \to \Gamma + \Delta\, \mathbf{1} \otimes \mathbf{1} \,,$$

effectively separates $\mathbf{1}$ from the degenerate subspace by shifting its eigenvalue from zero to Δ. Δ is chosen to be positive so that the eigenvalue is shifted into the vacant end of the eigenspectrum, where it is easily identified and reset to zero. However, Δ must not be so far separated from the negative eigenvalues that it causes the system to be ill conditioned. To avoid this, Δ should be on the same order as a "typical" eigenvalue. For most cases, a suitable shift should be the sign-inverted average eigenvalue. Since the trace of an $N \times N$ matrix is the sum of its N eigenvalues, Δ may be taken as

$$\Delta \equiv -N^{-1} \operatorname{Tr}(\Gamma) \,.$$

APPENDIX C: STABILITY OF SIGN STRUCTURES

Deuflhard et al. [23] describe an item's sign structure σ in terms of a cutoff ϵ, which determines if a component is sufficiently large so as to have a reliable sign or if the sign

should be instead represented as zero

$$\sigma(\overrightarrow{\boldsymbol{\psi}}_i, \epsilon) = (\sigma_1, \ldots, \sigma_m) \quad \text{with} \quad \sigma_j = \begin{cases} 0 & \text{if } |\boldsymbol{\psi}_j(i)| \leq \epsilon \\ \text{sign}(\boldsymbol{\psi}_j(i)) & \text{otherwise} \end{cases} .$$

The authors describe an iterative procedure for tuning this parameter so as to best decompose the eigenvectors into m partitions. Clustering is sensitive to this parameter since the sign structure may be used, in principle, to assign each item to a partition. The approximate solver uses sign structure in a much more restricted context: sign structures are used only to ensure that a candidate representative deemed to be the furthest item from the existing set of representatives belongs to a different cluster. That is, sign structures are used only for those items having eigenspace representations with large components. As such, the sign of a component is set to zero unless its magnitude is less than ϵ of the maximum amplitude in that eigenvector

$$\sigma(\overrightarrow{\boldsymbol{\psi}}_i, \epsilon) = (\sigma_1, \ldots, \sigma_m) \quad \text{with} \quad \sigma_j = \begin{cases} 0 & \text{if } |\boldsymbol{\psi}_j(i)| < \epsilon * \max_k |\boldsymbol{\psi}_j(k)| \\ \text{sign}(\boldsymbol{\psi}_j(i)) & \text{otherwise} \end{cases} .$$

ϵ is determined using the iterative algorithm similar to that described in Sec. 5 of Ref. [23].

---

[1] B. S. Everitt, S. Landau, and M. Leese, *Cluster Analysis* (Arnold, London, 2001).

[2] A. Ultsch, in *Workshop on Self-Organizing Maps* (Paris, 2005), pp. 75–82.

[3] A. Paccanaro, J. A. Casbon, and M. A. S. Saqi, Nucl. Acids Res. **34**, 1571 (2006).

[4] M. A. Wong and T. Lane, Journal of the Royal Statistical Society B **45**, 362 (1983).

[5] D. Korenblum and D. Shalloway, Phys. Rev. E **67** (2003), cited in appendix and available online at http://prola.aps.org/pdf/PRE/v67/i5/e056704.

[6] S. D. Kamvar, D. Klein, and C. D. Manning, in *Proc. International Joint Conference on Artificial Intelligence* (2003).

[7] A. Y. Ng, M. I. Jordan, and Y. Weiss, in *Proceedings of the 14th Neural Information Processing Systems Conference* (2002).

[8] C. J. Alpert and A. B. Kahng, in *ACM IEEE Design Automation Conference* (ACM Press, New York, NY, 1993), pp. 743–748.

[9] C. J. Alpert, A. B. Kahng, and S.-Z. Yao, Discrete Applied Mathematics **90**, 3 (1999).

[10] P. K. Chan, M. D. F. Schlag, and J. Y. Zien, in *ACM IEEE Design Automation Conference* (ACM Press, New York, NY, 1993), pp. 749–754.

[11] L. Hagen and A. Kahng, IEEE Trans. on CAD **11**, 1074 (1992).

[12] J. Shi and J. Malik, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (1997), pp. 731–737.

[13] S. T. Barnard and H. D. Simon, Concurrency: Practice and Experience **6**, 101 (1994).

[14] B. Hendrickson and R. Leland, SIAM J. Scientific Computing **16**, 452 (1995).

[15] E. R. Barnes, SIAM J. Alg. Disc. Meth. **3**, 541 (1982).

[16] M. R. Garey, D. S. Johnson, and L. Stockmeyer, in *Proc. ACM Symposium on Theory of Computing* (1974), pp. 47–63.

[17] M. Fiedler, Czechoslovak Mathematical Journal **25**, 619 (1975).

[18] B. Mohar, in *Graph Theory, Combinatorics, and Applications*, edited by Y. Alavi, G. Chartrand, O. R. Oellermann, and A. J. Schwenk (Wiley, 1991), vol. 2, pp. 871–898.

[19] N. Cristianini, J. Shawe-Taylor, and J. Kandola, in *Proc. Neural Information Processing Systems Conference* (2001).

[20] F. Rendl and H. Wolkowicz, Annals of Operations Research **58**, 155 (1995).

[21] K. M. Hall, Management Science **17**, 219 (1970).

[22] G. L. Scott and H. C. Longuet-Higgins, in *Proceedings of British Machine Vision Conference* (1990), pp. 103–108.

[23] P. Deuflhard, W. Huisinga, A. Fischer, and C. Schütte, Lin. Alg. Appl. **315**, 39 (2000).

[24] M. Meilă and J. Shi, in *Proc. International Workshop on Artificial Intelligence and Statistics* (2001).

[25] C. Schütte and W. Huisinga, Handbook of Numerical Analysis **X**, 699 (2003).

[26] L. Zelnik-Manor and P. Perona, in *Proc. Neural Information Processing Systems Conference* (2004).

[27] S. Guattery and G. L. Miller, in *Proc. ACM-SIAM Symposium on Discrete Algorithms* (SIAM, Philadelphia, PA, 1995), pp. 233–242.

[28] D. A. Spielman and S.-H. Teng, in *Proc. Annual Symposium on Foundations of Computer Science* (IEEE Computer Society, Washington, DC, 1996), pp. 96–105.

[29] H. D. Simon and S.-H. Teng, SIAM J. Scientific Computing **18**, 1436 (1997).

[30] R. Kannan, S. Vempala, and A. Vetta, in *Proc. Annual Symposium on Foundations of Com-*

puter Science (IEEE Computer Society, Washington, DC, 2000), pp. 367–377.

[31] M. Belkin and P. Niyogi, Neural Computation **15**, 1373 (2003).

[32] D. Harel and Y. Koren, in *Proc. Conference on Foundations of Software Technology and Theoretical Computer Science* (Springer-Verlag, London, UK, 2001), pp. 18–41.

[33] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis, in *Proc. Neural Information Processing Systems Conference* (2005).

[34] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis, Applied and Computational Harmonic Analysis **21**, 113 (2006).

[35] N. Tishby and N. Slonim, in *Proc. Neural Information Processing Systems Conference* (2000).

[36] L. Yen, D. Vanvyve, F. Wouters, F. Fouss, M. Verleysen, and M. Saerens, in *Proc. European Symposium on Artificial Neural Networks* (2005), pp. 317–324.

[37] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, Proc. Natl. Acad. Sci. USA **95**, 14863 (1998).

[38] W. Pentney and M. Meila, in *Proceedings of the National Conference on Artificial Intelligence* (2005), pp. 845–850.

[39] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, J. Mol. Biol. **247**, 536 (1995).

[40] R. Kubo, M. Toda, and N. Hashitsume, *Statistical Physics II: Nonequilibrium Statistical Mechanics* (Springer-Verlag, New York, 1985).

[41] J.-P. Ryckaert and A. Bellemans, Chem. Phys. Lett. **30**, 123 (1977).

[42] S. Chandrasekhar, Rev. Mod. Phys. **15**, 1 (1943).

[43] D. Shalloway, J. Chem. Phys. **105**, 9986 (1996).

[44] A. Ulitsky and D. Shalloway, J. Chem. Phys. **109**, 1670 (1998).

[45] M. Meilă and J. Shi, in *Proc. Neural Information Processing Systems Conference* (2000).

[46] G. W. Stewart, in *Mathematical Computer Performance and Reliability*, edited by G. Iazeolla, P. J. Courtois, and A. Hordijk (Elsevier, North Holland, 1984), pp. 287–302.

[47] Y. Weiss, in *Proceedings of IEEE International Conference on Computer Vision* (1999), pp. 975–982.

[48] G. W. Stewart, SIAM Review **15**, 727 (1973).

[49] S. F. Altschul, W. Gish, E. W. Meyers, and D. J. Lipman, J. Mol. Biol. **215**, 403 (1990).

[50] P. Pipenbacher, A. Schliep, S. Schneckener, A. Schönhuth, D. Schomburg, and R. Schrader, Bioinformatics **1**, 1 (2002).

[51] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, Nucl. Acids Res. **25**, 3389 (1997).

[52] A. J. Enright, S. Van Dongen, and C. A. Ouzounis, Bioinformatics **30**, 1575 (2002).

[53] A. J. Enright and C. A. Ouzounis, Bioinformatics **16**, 451 (2000).

[54] G. H. Golub and C. F. Van Loan, *Matrix Computations* (John Hopkins U. Press, Baltimore, Md., 1996), 3rd ed.

[55] D. A. Spielman and S. Teng, Journal of the ACM **51**, 385 (2004).

[56] D. C. Sorensen, SIAM J. Matrix Anal. Appl. **13**, 357 (1992).

[57] E. Anderson, Z. Bai, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide* (SIAM, Philadelphia, PA, 1999).

[58] I. Dhillon, Tech. Rep. UCB/CSD-97-971, UC Berkeley (1997).

[59] Z. Bai, J. Demmel, J. Dongarra, J. Langou, and J. Wang, in *CRC Handbook of Linear Algebra*, edited by L. Hogben (CRC Press, 2006), pp. 75–1–75–24.

[60] R. Lehoucq and D. Sorensen, in *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, edited by Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst (SIAM, Philadelphia, 2000).

[61] R. Lehoucq and D. Sorensen, in *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, edited by Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst (SIAM, Philadelphia, 2000).

[62] F. A. M. Gomes and D. C. Sorensen, Tech. Rep. TR97729, Rice University (1997).

[63] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods* (SIAM, Philadephia, 1998).

[64] A. Makhorin, *GNU linear programming kit: Reference manual http://www.gnu.org/software/glpk* (2006).

[65] C. Stimming, *Lapack++ (http://lapackpp.sourceforge.net)* (2006).

[66] D. Verma and M. Meilă, Tech. Rep. UW CSE 03-05-01, University of Washington (2003).

[67] A. Pothen, H. D. Simon, and K.-P. Liou, SIAM Journal on Matrix Analysis **11**, 430 (1990).

[68] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay, in *Proc. ACM-SIAM Symposium on Discrete Algorithms* (SIAM, Philadelphia, PA, 1999), pp. 291–299.

[69] J. D. Chodera, W. C. Swope, J. W. Pitera, and K. A. Dill, Multiscale Model. Simul. **5**, 1214

(2006).

[70] J. Kubelka, J. Hofrichter, and W. A. Eaton, Curr. Opin. Struct. Biol. **14**, 76 (2004).

[71] W. C. Swope, J. W. Pitera, and F. Suits, J. Phys. Chem. B **108**, 6571 (2004).

[72] N. Singhal, C. D. Snow, and V. S. Pande, J. Chem. Phys. **121**, 415 (2004).

[73] C. Schütte, A. Fischer, W. Huisinga, and P. Deuflhard, J. Comput. Phys. **151**, 146 (1999).

[74] Y. Sugita and Y. Okamoto, Chem. Phys. Lett. **314**, 141 (1999).

[75] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, Nucleic Acids Research **28**, 235 (2000).

[76] S. E. Brenner, C. Chothia, T. J. P. Hubbard, and A. G. Murzin, Methods in Enzymology **266**, 635 (1996).

[77] Y. Zhang and J. Skolnick, Proteins: Structure, Function, and Bioinformatics **57**, 702 (2004).